

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## SOFTWARE PRO MANUÁLNÍ OSTŘENÍ KAMERY S ROZLIŠENÍM 4K

SOFTWARE FOR MANUAL FOCUS OF CAMERA WITH 4K RESOLUTION

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Adam Sláma

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Kříž

BRNO 2019

# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Adam Sláma

**ID:** 174393

**Ročník:** 2

**Akademický rok:** 2018/19

## NÁZEV TÉMATU:

**Software pro manuální ostření kamery s rozlišením 4K**

## POKYNY PRO VYPRACOVÁNÍ:

Vyberte a optimalizujte otestované metody využívané k detekci ostrosti obrazu pro použití na sekvence snímků a vytvořte aplikaci pracující v reálném čase, která bude uživateli ukazovat míru ostrosti obrazu při manuálním ostření kamery s rozlišením 4K.

Doporučeným vývojovým nástrojem je MS Visual C++ a balíček knihoven OpenCV.

## DOPORUČENÁ LITERATURA:

[1] JOHNSON, B. Professional visual studio 2013: selected readings. 1st edition. Washington: Microsoft Press. Penguin education. ISBN 11-188-3204-3.

[2] JAN, J. Medical image processing, reconstruction and restoration: concepts and methods. Boca Raton: Taylor, 2006, 730 s. ISBN 08-247-5849-8.

**Termín zadání:** 1.2.2019

**Termín odevzdání:** 16.5.2019

**Vedoucí práce:** Ing. Petr Kříž

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Práce obsahuje analýzu v současné době používaných metod, jejichž cílem je určit míru ostrosti obrazu. Této analýzy bylo využito při návrhu programu, který vyhodnocuje míru ostrosti obrazu v procentuální míře, pracuje v reálném čase a spolupracuje s kamerou o rozlišení 4K a manuálně ostřicím objektivem. Dále aplikace dokáže ve snímku za určitých podmínek najít předem definovaný obrazec, který se využívá pro zefektivnění procesu ostření, a v tomto místě měřit míru ostrosti. Další možnost představuje metoda, která hledá nejvhodnější místo pro ostření ve střední části obrazu. Tyto metody a samotný program jsou v práci podrobně popsány. Závěrečná část práce obsahuje záznamy o testovacích měřeních a jejich výsledky.

## KLÍČOVÁ SLOVA

automatické ostření, ostření, DCT, histogram, vlnové koeficienty, hrany, DWT, koeficient zaostření, program, kamera, snímek, manuální ostření

## ABSTRACT

This Master thesis is focused on the analysis of currently used methods which whose target is to determine the rate of image focus. This analysis was used during the development of the program which evaluates the rate of image focus in percentage rate, works in real time and cooperates with a camera capable of 4k image resolution with a manual focus of the lenses. Application is then capable of a finding of a pre-defined image under certain circumstances which is being used for increasing of effectivity of image focusing. Another option is represented by a method that is searching the most suitable area for focusing in the center of the image. A detailed description of these methods and program itself are also included in the thesis. The final part of the thesis contains records of measurement tests with its results.

## KEYWORDS

autofocus, focus, DCT, histogram, wavelet coefficients, edge, DWT, coefficient of focus, program, camera, image, manual focus

SLÁMA, Adam. *Software pro manuální ostření kamery s rozlišením 4K*. Brno, 2019, 50 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Petr Kříž



## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Software pro manuální ostření kamery s rozlišením 4K“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Petru Křížovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora

# Obsah

<b>Úvod</b>	<b>10</b>
<b>1 Metody zaměřené na zaostření obrazu</b>	<b>11</b>
1.1 Fázová detekce . . . . .	11
1.2 Detekce kontrastu . . . . .	12
1.3 Laserová detekce . . . . .	13
1.4 Detektory využívající vlastnosti obrazu . . . . .	13
1.4.1 Detektor vlnových koeficientů . . . . .	14
1.4.2 Detektor šířky histogramu obrazu . . . . .	14
1.4.3 Detektor DCT . . . . .	15
1.4.4 Detektory hran . . . . .	15
1.4.5 Středofrekvenční DCT . . . . .	17
1.4.6 Funkce založená na korelačních opatřeních . . . . .	17
1.4.7 Normalizovaná odchylka . . . . .	18
1.4.8 Hledání největšího koeficientu zaostření . . . . .	18
<b>2 Metody zaměřené na hledání podobnosti v obraze</b>	<b>19</b>
2.1 Scale Invariant Feature Transform (SIFT) . . . . .	19
2.2 Speeded Up Robust Features (SURF) . . . . .	19
2.3 Oriented FAST and Rotated BRIEF (ORB) . . . . .	20
<b>3 Použitý software a hardware</b>	<b>21</b>
3.1 Software . . . . .	21
3.1.1 Microsoft Visual Studio 2015 . . . . .	21
3.1.2 Open Source Computer Vision Library (OpenCV) . . . . .	21
3.2 Hardware . . . . .	22
3.2.1 Kamera mvBlueFOX3-1100 . . . . .	22
3.2.2 Objektiv Yongnuo YN50mm F1.8 . . . . .	23
<b>4 Snímky pro analýzu metod k výpočtu ostrosti</b>	<b>24</b>
4.1 Parametry snímků . . . . .	24
<b>5 Analyzované metody</b>	<b>26</b>
5.1 Metoda s využitím sumy vlnových koeficientů . . . . .	26
5.2 Metoda s využitím DCT . . . . .	27
5.3 Metoda s využitím šířky histogramu . . . . .	28
5.4 Metoda s využitím normalizované odchylky . . . . .	29
5.5 Metoda s využitím detektorů hran I. . . . .	29

5.6	Metoda s využitím detektorů hran II. . . . .	29
<b>6</b>	<b>Porovnání výsledků analyzovaných metod</b>	<b>31</b>
<b>7</b>	<b>Program</b>	<b>34</b>
7.1	Obecný popis programu . . . . .	34
7.2	Stručný popis stěžejních částí programu . . . . .	35
7.2.1	Hledání obrazce ve scéně . . . . .	35
7.2.2	Výběr místa ostření bez obrazce . . . . .	36
7.2.3	Měření ostrosti . . . . .	37
<b>8</b>	<b>Testování funkčnosti programu</b>	<b>40</b>
8.1	Přednastavení programu . . . . .	40
8.2	Měření maximální míry ostrosti . . . . .	41
8.3	Testování vyhledávání obrazce a vyhledávání nejvhodnějšího místa pro ostření . . . . .	41
8.4	Zhodnocení výsledků . . . . .	42
<b>9</b>	<b>Závěr</b>	<b>43</b>
	<b>Literatura</b>	<b>44</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>47</b>
	<b>Seznam příloh</b>	<b>48</b>
<b>A</b>	<b>Návod k obsluze programu</b>	<b>49</b>
<b>B</b>	<b>Obsah přiloženého CD</b>	<b>50</b>

# Seznam obrázků

1.1	Princip fázového ostření. Vzdálenost čočky od snímáče: Daleko (1), ideálně (2) a blízko (3).	12
1.2	Ukázka principu laserového měření vzdálenosti pro výpočet správné polohy optiky.	13
1.3	Vlevo originální obraz, vpravo odpovídající dílčí pásma.	14
1.4	Ukázka histogramu obrázku číslo 9 - rozostřený obraz.	15
1.5	Ukázka obrazů po DCT: vlevo ostrý obraz, vpravo rozostřený.	16
3.1	Logo Microsoft Visual Studio.	21
3.2	Logo knihovny OpenCV.	22
4.1	Obrázek s maximální ostrotí vytvořený automatickým ostřením fotoaparátu.	25
4.2	Obrázek 6. pozice od maximálně zaostřeného snímku.	25
5.1	Ukázka výběru bloku na ostřeném objektu pro výpočet ostroty.	26
5.2	Blok rozložen DWT na bloky $W_{LH1}$ , $W_{HL1}$ , $W_{HH1}$ a $W_{LL1}$ .	27
5.3	Zobrazení histogramu u 3. snímku a 6. snímku.	28
5.4	Vlevo originální blok snímku, vpravo blok s implementovanou Laplacianovou maticí.	30
5.5	Vlevo konvoluce s maticí $L_Y$ , vpravo konvoluce s maticí $L_X$ .	30
6.1	Hodnoty časové náročnosti pro všechny otestované metody.	32
6.2	Normalizované hodnoty koeficientu zaostření otestovaných metod každého snímku.	33
7.1	Vývojový diagram programu.	35
7.2	Snímek kamery, který je doplněný o hodnotu ostroty, po zpracování programem.	36
7.3	Vývojový diagram metody určené pro vyhledání obrazce ve scéně.	37
7.4	Ukázka nalezení obrazce ve scéně.	38
7.5	Vývojový diagram metody určené pro vyhledání nejvhodnější oblasti na ostření ve středu obrazu.	38
7.6	Vývojový diagram metody určené pro vyhledání nejvhodnější oblasti na ostření ve středu obrazu.	39
7.7	Graf znázorňující vytvoření křivky aproximací bodů.	39
8.1	Obrazec, který je programem vyhledáván [19].	40

# Seznam tabulek

3.1	Základní parametry snímače kamery [15]. . . . .	23
3.2	Základní parametry objektivu kamery [16]. . . . .	23
4.1	Základní parametry fotoaparátu Sony [17]. . . . .	24
4.2	Parametry snímků. . . . .	24
5.1	Koeficienty s využitím součtu vlnových koeficientů podle vzorce 1.2. .	27
5.2	Koeficienty zaostření s využitím DCT. . . . .	28
5.3	Výsledné šířky histogramu (koeficienty zaostření) podle vzorce 1.3. . .	28
5.4	Koeficienty s využitím normalizované odchylky. . . . .	29
5.5	Hodnoty koeficientů vypočítané s pomocí Laplacianovi matice. . . . .	29
5.6	Hodnoty koeficientů vypočítané podle vzorce 1.5. . . . .	30
6.1	Normalizované hodnoty koeficientu ostrosti pro každý snímek v [%]. .	31
6.2	Hodnoty času, které odpovídají době výpočtů koeficientu zaostření u všech snímků. . . . .	32
8.1	Hodnoty koeficientů ostrosti získané z uložených snímků. . . . .	41

# Úvod

Tato diplomová práce se věnuje oblasti počítačového vidění, konkrétně vyhodnocování ostrosti obrazu snímaného kamerou. U automatického zaostřování ve fotoaparátech či mobilních telefonech jsou využívány různé metody ve spojitosti s měnicími se vlastnostmi obrazu v závislosti na ostrosti. Nejčastěji se setkáme s fázovým ostřením, s ostřením pomocí laseru nebo s ostřením využívajícím kontrast obrazu, detektory hran, diskretní kosinovou transformaci atd.

První část práce je zaměřená na analýzu různých druhů zaostřovacích metod obrazu a na následnou implementaci nejvýhodnějších metod do programu, který uživateli ukazuje míru ostrosti při manuálním ostření v reálném čase.

Analyzovány jsou metody s využitím diskretní kosinové transformace, diskretní vlnové transformace, detektorů hran, normalizované odchylky a šířky histogramu. Tyto metody se mezi sebou porovnají. Analýza metod na obrazu se provádí v programu MATLAB. Pro analýzu byla vytvořena sekvence fotografií s rozlišením 4K. Sekvence obsahuje 12 snímků pořízených s odlišnou vzdáleností čoček optiky.

Další část práce obsahuje popis výsledného programu, seznámení s použitými funkcemi knihovny OpenCV a popis zdrojového kódu doplněný vývojovými diagramy. Software se vyvíjí ve vývojovém prostředí Visual Studio 2015 a testuje se na kameře mvBlueFOX3 s objektivem Yongnuo. Program dokáže vyhledat v obraze předem definovaný objekt (do určité hodnoty zaostření), na kterém se má měřit ostrost, nebo si sám najde nejlepší oblast pro ostření uprostřed snímku, což bylo požadavkem vzhledem k budoucímu uplatnění kamery při zaostřování na velkou vzdálenost (100 a více metrů).

V závěru práce budou vyhodnoceny analyzované metody a zhodnocena funkčnost výsledného programu. Program může být využíván pro doostřování zejména průmyslových kamer, které nemají schopnost ovládat optiku a jsou závislé pouze na manuálním ostření. U 4K rozlišení lidské oko těžko odhadne, zda má snímek 100 % ostrost. K dosažení maximálního zaostření může pomoci právě zde popisovaný a vytvořený program.

# 1 Metody zaměřené na zaostření obrazu

Při vytváření obrazu fotoaparátem, kamerou nebo mobilním telefonem se posunem čoček v objektivu mění ostrost výsledného obrazu. Rozmístění čoček určuje pomyslnou rovinu zaostření neboli hloubku ostrosti. Objekty ležící v této rovině se zobrazují ostře, objektům, které se nacházejí před či za pomyslnou rovinou, ostrost se vzdáleností od roviny klesá.

Dnešní technologie určené pro zachycování obrazu využívají z velké části technologii automatického zaostřování (autofokus). Tato technologie využívá výpočetní metody určené k nalezení určité pozice čoček, v nichž má daný objekt v obraze požadovanou ostrost. Metody využívají parametry snímků, které se mění se vzdáleností čoček, z nichž následně vypočítávají koeficient ostrosti. Poloha čoček, při které bude koeficient ostrosti s největší hodnotou, se stává tou neoptimálnější a obraz by měl mít největší možnou ostrost.

V současné době se u autofokusu nejvíce používají metody pro detekci kontrastu (mobilní telefony) a metody pro detekci fází (vyšší třídy telefonů). Obě metody mají uplatnění i u moderních zrcadlových fotoaparátů. Dále se používá aktivní autofocus, a to např. laserový. Jeho vyšší rychlost výpočtu a orientace v horších podmínkách potřebuje ke své funkčnosti další hardware, což zvyšuje celkovou cenu zařízení.

V této práci budou některé z výše zmíněných metod otestovány. Jedná se o metody pracující s kontrastem obrazu, s detektory hran (dva odlišné postupy), s diskrétní kosinovou transformací, s normalizovanou odchylkou a se součtem vlnových koeficientů [1, 2].

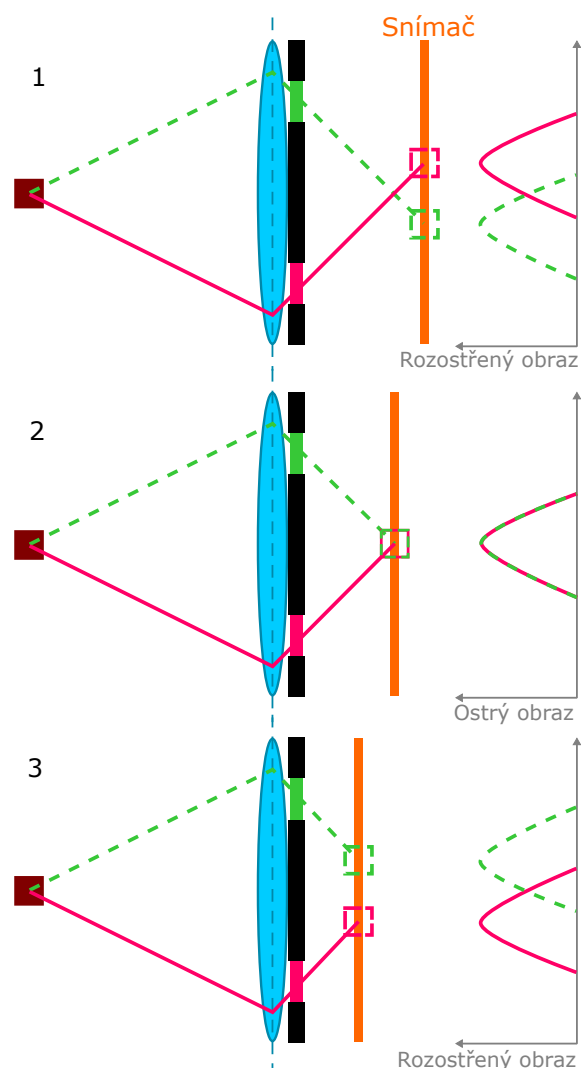
## 1.1 Fázová detekce

Fázová detekce se řadí mezi nejnovější a nejpokročilejší metody zaostřování obrazu. Nejběžněji se používá u digitálních zrcadlovek, avšak v poslední době se tato technologie začíná objevovat i u špičkových telefonů, zejména značek Apple a Samsung.

Metoda spočívá v porovnávání světla ze dvou otvorů na protilehlých stranách čočky. Každý z otvorů má svůj snímač. Pokud je obraz správně zaostřen, tak se světelné paprsky objevují přímo ve středu snímače, tzn. že snímače mají identické obrazy. V opačném případě, kdy se objekt nachází mimo ohnisko, se paprsky od středu vzdalují, tzn. že obrazy na snímačích nejsou identické, jak lze vidět na obrázku 1.1 níže. Pro zvětšení přesnosti se používá větší množství protilehlých otvorů.

Z grafů ve zmíněném obrázku je patrné, že fázový rozdíl se dá použít jak na určení směru pro doostření tak i pro odhad optimální vzdálenosti. Jedná se tedy o rychlou a přesnou metodu [1].





Obr. 1.1: Princip fázového ostření. Vzdálenost čočky od snímače: Daleko (1), ideálně (2) a blízko (3).

## 1.2 Detekce kontrastu

U ostření s detekcí kontrastu se vychází z toho, že při maximálně zaostřeném objektu má obraz největší kontrast a při rozostřování se kontrast začne snižovat. Když je obraz mimo ostrost, tak lze předpokládat, že barvy budou mít tendenci se vzájemně rozmazávat. Z toho plyne nižší úroveň celkového kontrastu.

Zařízení tedy vypočítá hodnotu kontrastu s určitou pozicí čoček. Poté změní rozestavení optické soustavy a změří znovu hodnotu kontrastu. Srovná vypočítané hodnoty a vyhodnotí, kterým směrem kontrast roste. Takovými kroky se postupně dostane až k maximálnímu kontrastu, tzn. nejlépe zaostřenému obrazu. Tuto metodu dnes využívá většina chytrých telefonů.

Velkou výhodou tohoto systému je, že nepotřebuje další hardware, ale kvůli

porovnávání a výpočtu hodnot kontrastu z každé pozice pracuje delší dobu než jiné metody. Přesto se řadí k metodám s dobrými výsledky [2].

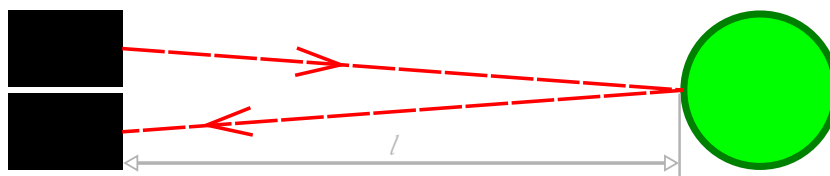
### 1.3 Laserová detekce

Metody s detekcí kontrastu a s detekcí fází analyzují světlo přicházející do optiky. Tyto metody se označují jako pasivní automatické ostření. Laserová detekce se označuje jako aktivní ostření, jelikož se nespolehá na příchozí světlo. Tudíž se jedná o velmi odlišný systém, který z vlastního zdroje vyšle infračervené světlo k zaostřovanému objektu. Od vyslání světla po příjem se měří čas, z něhož se vypočítá vzdálenost od objektu  $l$ . K výpočtu slouží jednoduchý vzorec

$$l = \frac{c \cdot t}{2}, \quad (1.1)$$

kde  $c$  je rychlost světla a  $t$  je čas, za jaký se světlo vrátí zpět do přijímače.

Pomocí vypočítané vzdálenosti se optika posune do příslušné polohy. Výhodou této metody je, že pracuje i při horších světelných podmínkách, jelikož používá vlastní světelný zdroj. Avšak je použitelná jen u kratších vzdáleností, protože u větších (cca 5 m) dochází k chybnosti ostření následkem špatných odrazů světla apod. Dále může způsobovat problémy ostření přes plot či sklo. Metoda potřebuje přídavný hardware, což zvyšuje i cenu zařízení.



Obr. 1.2: Ukázka principu laserového měření vzdálenosti pro výpočet správné polohy optiky.

### 1.4 Detektory využívající vlastnosti obrazu

Výše zmíněné metody jsou v této době nejčastěji využívány. My budeme k analyzování přistupovat podobně jako je tomu u detekce kontrastu. Nebudeme používat žádný přídavný hardware, ale budeme vycházet pouze z obrazových vlastností, které se změnou ostrosti mění.

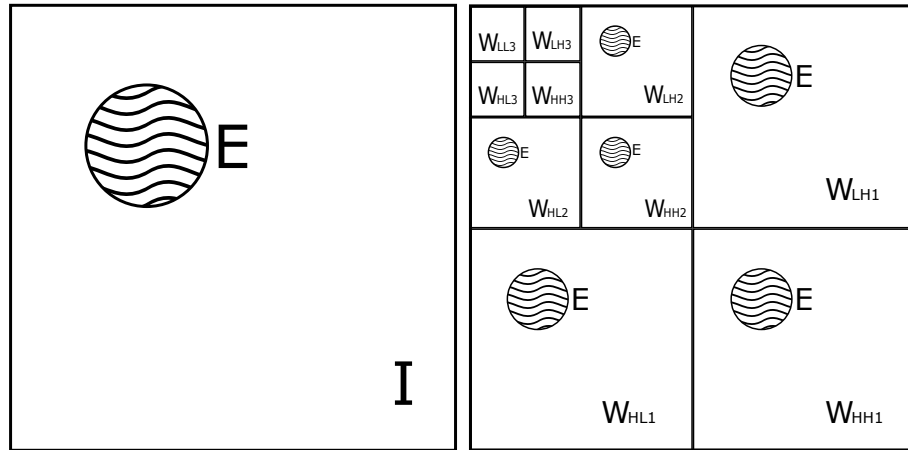
### 1.4.1 Detektor vlnových koeficientů

Metody založené na měření vlnových délek se nejčastěji zakládají na statických vlastnostech koeficientů diskrétní vlnové transformace (DWT). Na první úrovni DWT je obraz  $I$  rozložen do čtyř dílčích obrazů  $W_{LH1}$ ,  $W_{HL1}$ ,  $W_{HH1}$  a  $W_{LL1}$  jako je znázorněno na obrázku 1.3. První tři koeficienty označují tři detailní pásma a čtvrtý označuje pásmo hrubé aproximace. Pro vyšší úroveň DWT je hrubá aproximace následně rozložena do detailních a hrubých dílčích pásem. Informace o podrobnostech a hrubých dílčích pásmech se pak použije pro kompenzaci fokusu.

Yang a Nelson [3] navrhli vzorec pro vyhodnocování ostroty z dílčích pásem

$$\phi = \sum_{(i,j) \in \Omega_D} |W_{LH1}(i,j)| + |W_{HL1}(i,j)| + |W_{HH1}(i,j)|, \quad (1.2)$$

kde  $\Omega_D$  je odpovídající okno  $\Omega$  v dílčích pásmech DWT. Jelikož se koeficient bude počítat podle všech vlnových operátorů, může se vyloučit sousedství každého dílčího pásma. Tudíž je  $\Omega_D$  stejný jako  $\Omega$  [4].



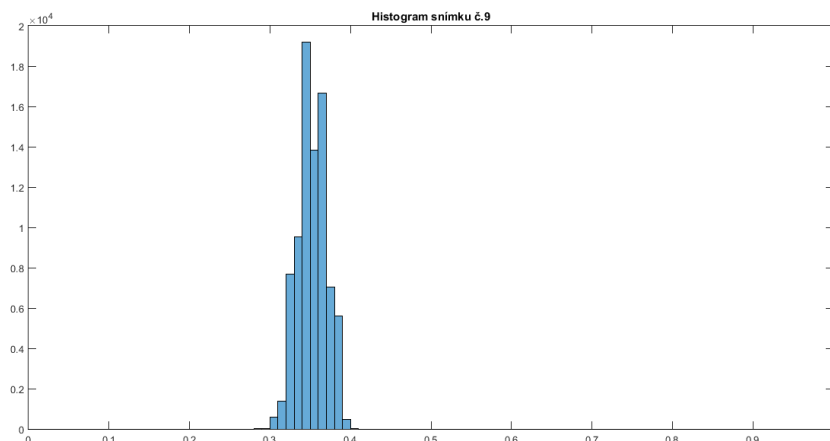
Obr. 1.3: Vlevo originální obraz, vpravo odpovídající dílčí pásma.

### 1.4.2 Detektor šířky histogramu obrazu

Pro vyhodnocení ostroty lze využít i šířku obrazového histogramu. Tu vypočítáme vzorcem

$$\phi = \max(k|H > 0) - \min(k|H > 0), \quad (1.3)$$

kde se odečte maximální nenulová hodnota histogramu  $H$  s minimální nenulovou hodnotou. Předpokládá se, že s rostoucí šířkou histogramu roste i ostrost obrazu [4].



Obr. 1.4: Ukázka histogramu obrázku číslo 9 - rozostřený obraz.

## Histogram obrazu

Histogram je sloupcový graf. Sloupce mají stejnou šířku. Pozice sloupce vyjadřuje danou hodnotu z určitého intervalu a výška sloupce udává četnost dané hodnoty.

U digitálních obrazů sloupce histogramu vyjadřují jednotlivé úrovně jasu od černé po bílou. Jedná se tedy o grafické znázornění jasového rozdělení obrazu. Pomocí histogramu lze objektivně určit jasové vlastnosti obrazu jako jsou kontrast, expozice apod.

### 1.4.3 Detektor DCT

Diskrétní kosinová transformace se podobá Fourierově transformaci. Základní funkcí je převod časového obrazu na frekvenční spektrum. Na rozdíl od DFT se výsledné hodnoty pohybují pouze v reálných koeficientech. Výhodou je, že se energie soustředí do nižších frekvencí, což se využívá u kompresních algoritmů.

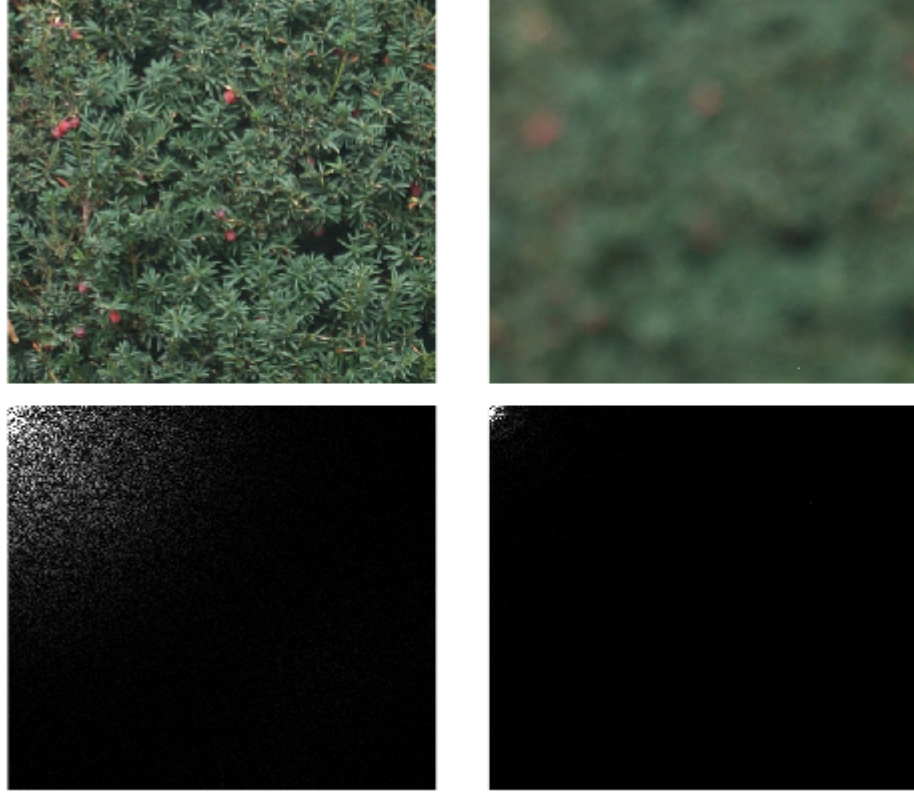
První prvek udává střední hodnotu celého transformovaného signálu a zbylé prvky udávají koeficienty postupně rostoucích frekvencí. Vyšší frekvence se objevují s přibývajícím ostrostí snímku. Při maximálním zaostření je frekvenční spektrum obsazeno nejvyššími frekvencemi.

Obrázek 1.5 dokazuje, že je veškerá energie obrazu mířená do levého horního rohu. Také dokazuje, že rozostřený obraz neobsahuje tak velké frekvence jako je tomu u ostrého snímku. A toho lze využít při vyhodnocování ostrosti obrazu.

### 1.4.4 Detektory hran

#### Energie laplasiánů

Pro koeficient ostrosti je použita energie druhé derivace obrazu



Obr. 1.5: Ukázka obrazů po DCT: vlevo ostrý obraz, vpravo rozostřený.

$$\phi = \sum_{(i,j) \in \Omega(x,y)} \Delta I(i,j)^2, \quad (1.4)$$

kde  $\Delta I$  je obrazem získaným spojením  $I$  a Laplacianovou maskou [4].

### Modifikovaný laplasian

Nayar [5] navrhl metodu na základě alternativní definice Laplasianu

$$\phi = \sum_{(i,j) \in \Omega(x,y)} \Delta_m I(i,j), \quad (1.5)$$

kde  $\Delta_m I$  se rovná modifikovanému Laplacianu z obrazu  $I$ . Jeho výpočet zní

$$\Delta_m I = |I \otimes L_X| + |I \otimes L_Y|. \quad (1.6)$$

$$L_X = \begin{bmatrix} -1 & 2 & -1 \end{bmatrix} \quad (1.7)$$

a

$$L_Y = L_X^T = \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix} \quad (1.8)$$

jsou konvoluční masky určené pro výpočet modifikovaných Laplacianů [4].

### 1.4.5 Středofrekvenční DCT

Autoři Sang-Yong Lee, Kumar, Ji-Man Cho, Sang-Won Lee, Soo-Won Kim [6] analyzovali účinek koeficientů DCT pásmového průchodu na měření zaostření. Stejní autoři navrhli  $4 \times 4$  operátor DCT pro extrakci centrálního koeficientu  $c(4, 4)$ , který se používá jako měření zaostření. Tento operátor lze vypočítat jako:

$$\phi = \sum_{i,j} (g(i, j) \otimes O)^2, \quad (1.9)$$

kde  $g(i, j)$  je hodnota DCT koeficientu  $g$  nacházejícího se na souřadnicích  $i, j$  a

$$O = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}. \quad (1.10)$$

Autoři Mateos-Pérez, Redondo, Nava [7] a kolektiv metodu doporučují pro aplikace pracující v reálném čase, pro malou výpočetní náročnost a pro dobré výsledky ostroty.

### 1.4.6 Funkce založená na korelačních opatřeních

Na funkci korelačního opatření byly navrženy dvě zaostřovací metody s velmi dobrým výkonem za přítomnosti hluku. První je založená na autokorelační funkci a druhá na funkci standardní odchylky. První metoda zní

$$\phi = \sum_{i=1}^{M-1} \sum_{j=1}^N g(i, j) \cdot g(i+1, j) - \sum_{i=1}^{M-2} \sum_{j=1}^N g(i, j) \cdot g(i+2, j), \quad (1.11)$$

kde  $g(i, j)$  je šedá úroveň pixelu  $g$  nacházejícího se na souřadnicích  $i, j$  a  $N$  a  $M$  jsou rozměry obrazu. Druhá metoda zní

$$\phi = \sum_{i=1}^{M-1} \sum_{j=1}^N g(i, j) \cdot g(i+1, j) - MN\bar{g}^2, \quad (1.12)$$

kde  $\bar{g}$  je střední hodnotou pixelu v obraze [7, 8].

Mateos-Pérez, Renondo a Nava [7] doporučují první z metod pro aplikace pracující v reálném čase.

### 1.4.7 Normalizovaná odchylka

Tato metoda kompenzuje rozdíly v průměrném jasu obrazu u různých obrazů [8]:

$$\phi = \frac{1}{MN\bar{g}} \sum_M \sum_N |g(i, j) - \bar{g}|^2, \quad (1.13)$$

kde  $g(i, j)$  je šedá úroveň pixelu,  $\bar{g}$  je střední hodnotou pixelu v obraze a  $N$  a  $M$  jsou rozměry obrazu v pixelech.

### 1.4.8 Hledání největšího koeficientu zaostření

Po získání zaostřovacích koeficientů z různých ohniskových vzdáleností přichází další bod, a tím je nalezení nejvyššího možného zaostřovacího koeficientu. Z již nalezených koeficientů se může např. interpolací či jiným způsobem vytvořit křivka. Nejvyšší bod křivky by nám udával hodnotu zaostřovacího koeficientu, kterou by měl následovat nejlépe zaostřený obraz viz obrázek 7.7.

## 2 Metody zaměřené na hledání podobnosti v obraze

Pro automatické nalezení objektu v obraze, na který se bude následně zaostřovat, se využívají metody určené k hledání podobnosti v obrazech. Tyto metody v obraze hledají tzv. klíčové (důležité) body, které je detektor schopný najít opakovaně i za jiných zobrazovacích podmínek. Mezi nejznámější patří SIFT, SURF a ORB.

### 2.1 Scale Invariant Feature Transform (SIFT)

Algoritmus SIFT byl navržen Davidem Lowem. Tento algoritmus zvládá řešit rotaci, transformaci, měřítko obrazu a různé úhly (do určité míry), ve kterých byly snímky pořízeny. Metoda se skládá ze čtyř základních kroků.

V prvním kroku se odhaduje poloha lokálních extrémů a prostorová velikost měřítka z rozdílu obrazů vyhlazených pomocí Gaussovy funkce DoG. Výhodou je velmi malá výpočetní náročnost, jelikož se od sebe sousední obrazy pouze odečítají.

Druhý krok se věnuje zpřesnění lokalizace a porovnání stability každého bodu. Odstraněny mohou být nestabilní body s nízkým kontrastem, tedy ty, které jsou citlivé na šum. Dále se odstraní i body, které byly lokalizovány podél hran a to použitím Hassovy matice.

Ve třetím kroku metody SIFT se přiřadí orientace klíčového bodu na základě gradientu lokálního obrazu. Tímto metoda dosahuje nezávislosti na rotaci. Pro nezávislost na měřítku se bodům získaným postupem popsaným výše přiřadí vygenerované měřítko, které je bodu nejbližší.

A v posledním kroku se tvoří deskriptory. Výpočet probíhá na základě velikosti a orientace gradientu obrazu. Tímto by mělo dojít k nezávislosti na místě pořízení snímku 3D prostoru a k nezávislosti na změně osvětlení [9, 10].

### 2.2 Speeded Up Robust Features (SURF)

U této metody už název napovídá, že její předností je rychlost. Pro zvýšení rychlosti procesu se snížila výpočetní náročnost díky využití integrálního obrazu při konvolucích. Dále se zde využívají znalosti z předešlého detektoru (využití Hessovy matice, formování deskriptorů), kde je snaha co nejvíce je zjednodušit. Aproximace derivací Gaussových funkcí, které mohou být díky integrálnímu obrazu spočítány rychleji, také vedou ke snížení celkového času výpočtu.

Pro vyhledávání významných bodů se zde využívá také Hessova matice. Ta má tvar



$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \quad (2.1)$$

kde  $\mathbf{x} = (x, y)$  je daným bodem v obraze o měřítku  $\sigma$ . Konvoluci druhé derivace Gaussovy funkce s obrazem v bodě  $\mathbf{x}$  značí  $L_{xx}(\mathbf{x}, \sigma)$ ,  $L_{xy}(\mathbf{x}, \sigma)$  a  $L_{yy}(\mathbf{x}, \sigma)$ .

Okolí kolem klíčového bodu se rozdělí do subregionů a poté se pro každou oblast využije filtr Haarových vln. Z toho se získá deskriptor funkce SURF [10, 11].

## 2.3 Oriented FAST and Rotated BRIEF (ORB)

Metoda ORB se skládá z detektoru klíčových bodů FAST a deskriptoru BRIEF s některými modifikacemi. Na začátku se najdou klíčové body pomocí FAST. Poté se pomocí Harrisova rohového měřítka naleznou  $N$  nejlepších bodů. Následně se okolo rohu vypočítá těžiště z intenzity oblasti. Směr vektoru tohoto rohového bodu udává orientaci. Tím se zvětší odolnost vůči rotaci. Deskriptor BRIEF má špatné vlastnosti při rotování oblasti, tudíž se počítá rotační matice z orientace bodů. Pak se deskriptor řídí podle vypočítané orientace [10, 12].

## 3 Použitý software a hardware

V této části práce se pojednává o vývojovém prostředí, ve kterém samotný program vznikne, a o hardwaru, na kterém bude program testován.

### 3.1 Software

Tato diplomová práce využívá k vytvoření finálního kódu vývojové prostředí Microsoft Visual Studio 2015. Důležitou částí je doplnění vývojového prostředí o balíček knihoven OpenCV a SDK (Software Development Kit) průmyslových kamer Matrix Vision.

Samotný kód je psán programovacím jazykem C++. Tedy jedním z nejrozšířenějších jazyků dnešní doby. Mezi jeho přednosti se řadí podpora objektově orientovaného, procedurálního a generického programování.

#### 3.1.1 Microsoft Visual Studio 2015

Microsoft Visual Studio vytvořila firma Microsoft Corporation a jedním z nejpoužívanějších vývojových prostředí. Podporuje platformy jako OS Windows, kde v jednom vývojovém prostředí lze vytvářet program jak pro Windows, Android, tak i iOS. Může se jednat o desktopové a webové aplikace nebo zařízení.

Díky integrovanému debuggeru pracuje Microsoft Visual Studio na úrovni kódu i stroje. Dále obsahuje designer formulářů pro tvorbu aplikací GUI, designer webu, tříd a databázových schémat i editor kódu. C/C++, C# a VB.NET jsou vestavěné jazyky. Další jazyky mohou být doinstalovány pomocí jazykové služby [13].



Obr. 3.1: Logo Microsoft Visual Studio.

#### 3.1.2 Open Source Computer Vision Library (OpenCV)

OpenCV je open source knihovnou určenou pro počítačové vidění a strojové učení. Je navržena tak, aby poskytovala společnou infrastrukturu pro aplikace počítačového

vidění a urychlila strojové vnímání u komerčních produktů. Jedná se o produkt s licenci BSD, tzn. usnadňuje podnikům upravovat a používat kód.

V knihovně najdeme více než 2500 optimalizovaných algoritmů. Zahrnuje komplexní sadu klasických i moderních funkcí počítačového vidění a algoritmy strojového učení. Tyto algoritmy mohou být použity k detekci a rozpoznání tváří, identifikaci objektů, klasifikaci lidských činností ve videích, sledování pohybu kamer, sledování pohybujících se objektů, extrahování 3D modelů objektů, vytváření 3D bodových mraků ze stereofonních kamer, spojování obrazů a vytváření vysokého rozlišení. Obraz celé scény, najít podobné snímky z obrazové databáze, odstranit červené oči ze snímků pořízených pomocí blesku, sledovat pohyby očí, rozpoznat scénérii a vytvořit značky, které ji překryjí rozšířenou realitou atd. OpenCV má více než 47 tisíc uživatelů a odhadovaný počet stažení přesahuje 18 milionů. Knihovna je hojně využívána ve firmách, výzkumných skupinách a vládních orgánech.

Knihovna má rozhraní C++, Python, Java a MATLAB a podporuje Windows, Linux, Android a Mac OS. OpenCV se opírá především o aplikace v reálném čase. Aktivně se vyvíjí plně funkční rozhraní CUDA a OpenCL. Existuje více než 500 algoritmů a asi desetkrát více funkcí, které tyto algoritmy skládají nebo podporují. OpenCV je napsán nativně v C++ [14].



Obr. 3.2: Logo knihovny OpenCV.

## 3.2 Hardware

Kamera mvBlueFOX3 s objektivem Yongnuo se stanou testovacím hardwarem pro výsledný program. V této části bude zmínka o jejich základních parametrech. Program se bude používat na notebooku Toshiba Satellite L855-10W s procesorem Intel Core i7-3610QM, 12 GB RAM a operačním systémem Windows 10 Home.

### 3.2.1 Kamera mvBlueFOX3-1100

Pro přehlednost jsou v tabulce 3.1 uvedeny technické parametry snímače kamery. Kamera od společnosti Matrix Vision má rozlišení  $3856 \times 2764$ , a tím splňuje poža-

davek na 4K rozlišení. Při maximálním rozlišení dokáže kamera vyprodukovat až 7 snímků za sekundu. Použitý typ senzoru je CMOS.

Tab. 3.1: Základní parametry snímače kamery [15].

Dodavatel	Aptina
Název	MT9J003
Rozlišení	$3856 \times 2764$
Velikost	1/2,3"
Max. FPS	7,3
Typ	CMOS

### 3.2.2 Objektiv Yongnuo YN50mm F1.8

Výše uvedená kamera bude vybavena objektivem YN50mm F1.8 od firmy Yongnuo. Ohnisková vzdálenost, zorný úhel, zvětšení a další parametry jsou uvedeny v tabulce 3.2.

Tab. 3.2: Základní parametry objektivu kamery [16].

Ohnisko	50 mm
Rozsah clony	f/1,8 až f/22
Zorný úhel	46°
Zvětšení	0,15
Min. zaostřovací vzdálenost	45 cm

## 4 Snímky pro analýzu metod k výpočtu ostrosti

Důležitým bodem této práce je vytvoření snímků, dle kterých se budou vyhodnocovat a porovnávat výsledky metod. Tyto výsledky nám určí, která ze zkoumaných metod se následně použije pro software manuálního ostření.

### 4.1 Parametry snímků

Snímky byly tvořeny fotoaparátem Sony Alpha DSLR-A350 (viz tabulka 4.1 níže). S pomocí stativu byla zhotovena sekvence 12 snímků s totožnou scénou. Pomocí autofokusu fotoaparátu byla nalezena neoptimálnější pozice čoček a byl vytvořen snímek s největší ostroší objektu (viz obrázek 4.1). Z této pozice se po malých krocích manuálním posuvem měnila pozice čoček, a tím i hloubka ostrosti. Každý krok reprezentuje zaznamenaný snímek.

Tab. 4.1: Základní parametry fotoaparátu Sony [17].

Název	Sony Alpha DSLR-A350
Typ fotoaparátu	Jednooká digitální zrcadlovka
Typ senzoru	$23,6 \times 15,8$ mm CCD (3:2)
Výstupní formáty	JPEG, RAW, RAW+JPEG, 16:9 volitelně
Max. rozlišení snímku	$4592 \times 3056$ px (3:2)
Rozlišení celkové/efektivní	14,9/14,2 MPix

Jelikož má výsledný software pracovat s rozlišením 4K, byly snímky vytvořeny s rozlišením  $4592 \times 3056$  pixelů. Další parametry snímků obsahuje tabulka 4.2.

Tab. 4.2: Parametry snímků.

Rozměry	$4592 \times 3056$	[pixelů]
Horizontální rozlišení	72	[dpi]
Vertikální rozlišení	72	[dpi]
Bitová hloubka	24	[b]
Barevné rozlišení	sRGB	



Obr. 4.1: Obrázek s maximální ostroostí vytvořený automatickým ostřením fotoaparátu.



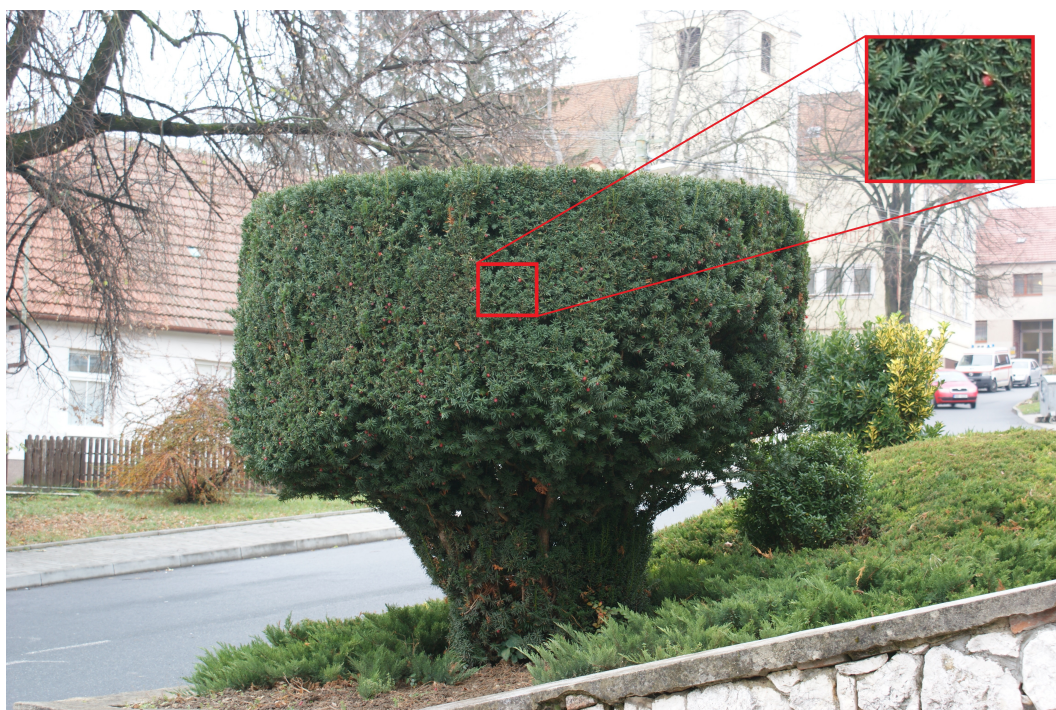
Obr. 4.2: Obrázek 6. pozice od maximálně zaostřeného snímku.



## 5 Analyzované metody

Vzhledem k tomu, že nemáme k dispozici jiný hardware než snímač obrazu (na rozdíl od laserových nebo fázových detektorů ostrosti), budou následující metody vycházet pouze z vlastností obrazu podobně jako kontrastní detektor ostrosti. Pro analýzu byly zvoleny metody s využitím DCT, detektorů hran, šířky histogramu, normalizované odchylky a sumy vlnových koeficientů.

Pro zmenšení výpočetní náročnosti u tak velkých snímků vybereme pouze malý blok, ve kterém se nachází ostřený objekt (viz obrázek 5.1). Jelikož máme statickou scénu, můžeme si dovolit nechat velikost a pozici bloku u všech snímků stejnou. Pro kvalitní porovnání výsledků necháme všechny metody pracovat s bloky o stejné velikosti a se stejným umístěním.



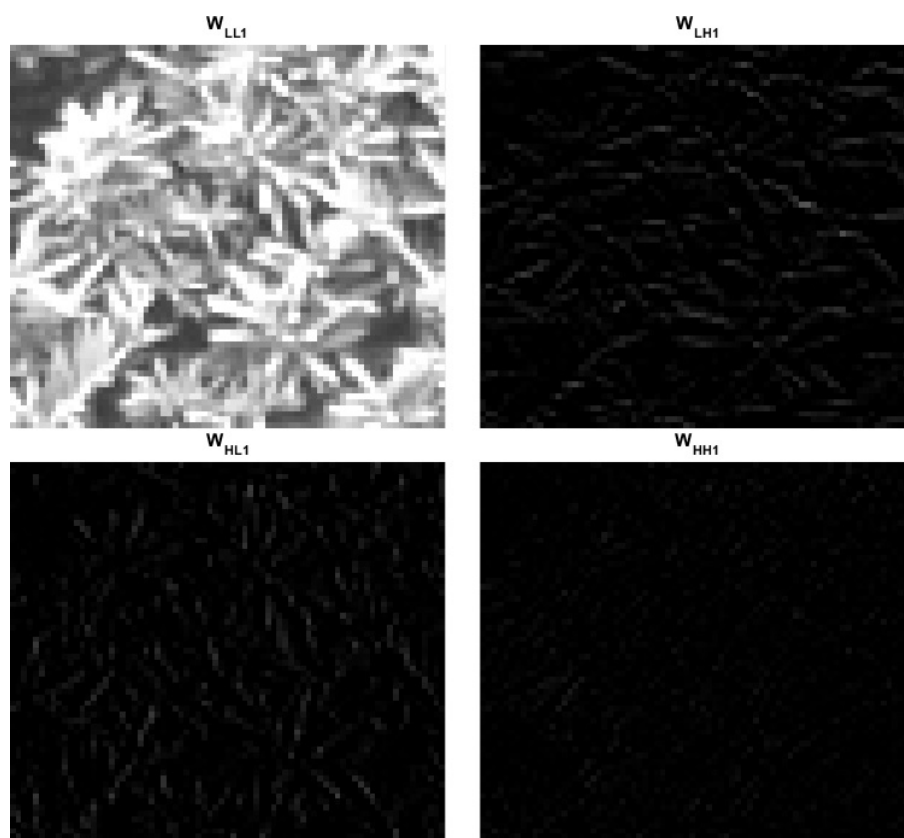
Obr. 5.1: Ukázka výběru bloku na ostřeném objektu pro výpočet ostrosti.

Všechny metody pracují s šedotónovým obrazem. Proto se bloky převedou z RGB do odstínů šedi. Tímto je připravený materiál pro následující metody.

### 5.1 Metoda s využitím sumy vlnových koeficientů

Pomocí první úrovně DWT byl obrazový blok rozdělen na čtyři obrazy  $W_{LH1}$ ,  $W_{HL1}$ ,  $W_{HH1}$  a  $W_{LL1}$  (viz obrázek 5.2). Hodnoty bloků  $W_{LH1}$ ,  $W_{HL1}$  a  $W_{HH1}$  se podle vzorce 1.2 pomocí absolutní hodnoty převedly na kladná čísla, která se následně sečetla.

U každého bloku byl tímto postupem vypočítán koeficient pro porovnání ostrosti



Obr. 5.2: Blok rozložen DWT na bloky  $W_{LH1}$ ,  $W_{HL1}$ ,  $W_{HH1}$  a  $W_{LL1}$ .

a hodnoty byly zaneseny do tabulky 5.1. Zpracování a výpočet koeficientů touto metodou pro všechny bloky trvalo 0,930838 sekundy.

Tab. 5.1: Koeficienty s využitím součtu vlnových koeficientů podle vzorce 1.2.

<b>Snímek</b>	<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>
Koeficient zaostření [Ø]	40,7	113,4	<b>456,2</b>	95,8	42,9	35,1
<b>Snímek</b>	<b>7.</b>	<b>8.</b>	<b>9.</b>	<b>10.</b>	<b>11.</b>	<b>12.</b>
Koeficient zaostření [Ø]	35,2	34,7	35,3	34,1	34,5	35

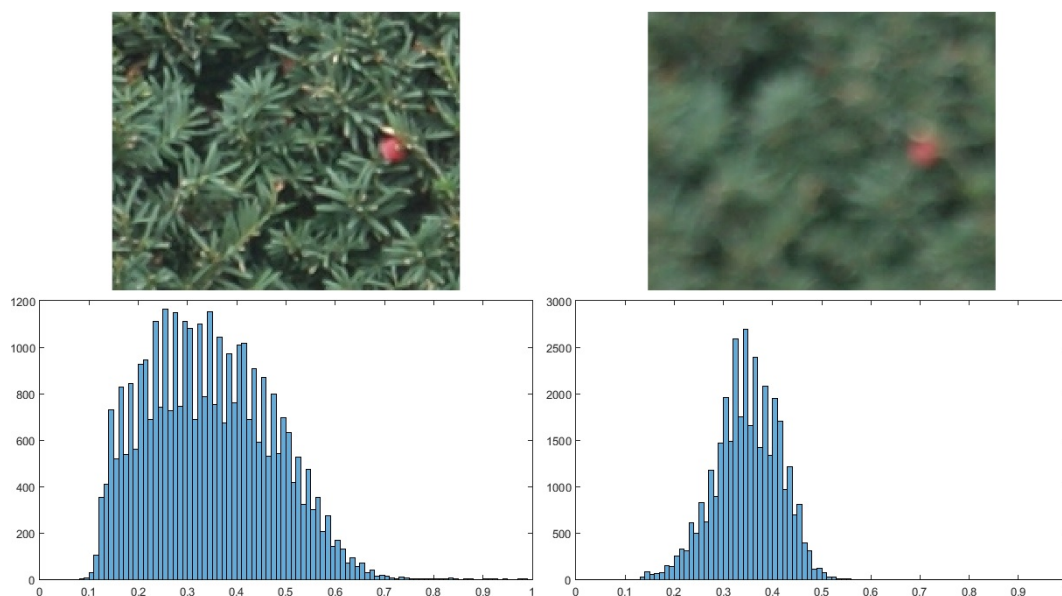
## 5.2 Metoda s využitím DCT

Jak bylo výše zmíněno, blok převedený do odstínů šedi se pomocí DCT převedl do frekvenční oblasti. Hodnoty pomocí absolutní hodnoty byly převedeny na kladná čísla a sečteny. Součet frekvenčních složek nám posloužil jako koeficient zaostření, kde opět nejvyšší součtová hodnota určuje nejostřejší obraz. Čas pro vypočítání koeficientů zaostření touto metodou pro všechny obrazové bloky byl 0,015587 sekundy.



Tab. 5.2: Koeficienty zaostření s využitím DCT.

Snímek	1.	2.	3.	4.	5.	6.
Koeficient zaostření [Ø]	138,9	307,2	<b>981,1</b>	263,7	144,4	119,5
Snímek	7.	8.	9.	10.	11.	12.
Koeficient zaostření [Ø]	115,6	112,8	109,1	106,2	106,1	106,7



Obr. 5.3: Zobrazení histogramu u 3. snímku a 6. snímku.

### 5.3 Metoda s využitím šířky histogramu

Tato metoda pracuje na jednoduchém principu. Z bloku se vytvoří histogram a vypočítá se jeho šířka odečtením minimálního nenulového bodu od maximálního nenulového bodu.

Změnu histogramu znázorňuje obrázek 5.3. Se snižující se ostroty se snižuje i šířka histogramu. Výsledné hodnoty koeficientů zaostření znázorňuje tabulka 5.3. Program touto metodou vypočítal koeficienty zaostření všech bloků za  $0,676 \cdot 10^{-3}$  s.

Tab. 5.3: Výsledné šířky histogramu (koeficienty zaostření) podle vzorce 1.3.

Snímek	1.	2.	3.	4.	5.	6.
Koeficient zaostření [Ø]	0,1689	0,5216	<b>0,8980</b>	0,4191	0,1961	0,1137
Snímek	7.	8.	9.	10.	11.	12.
Koeficient zaostření [Ø]	0,0902	0,0667	0,0549	0,0549	0,0471	0,0510

## 5.4 Metoda s využitím normalizované odchylky

Metoda byla otestována podle vzorce 1.13. Z daného šedotónového bloku se vypočítala střední hodnota  $\bar{g}$ . Ta se odečetla od hodnoty každého pixelu a výsledek se převedl na kladná čísla pomocí absolutní hodnoty a převedl na druhou mocninu. Takto matematicky upravené hodnoty pixelů se sečetly a podělily součinem střední hodnoty  $\bar{g}$  a rozměry obrazu  $N$  a  $M$ . Výsledné hodnoty každého obrazu obsahuje tabulka 5.4. Za 0,064672 sekundy program touto metodou vypočítal koeficienty zaostření všech bloků.

Tab. 5.4: Koeficienty s využitím normalizované odchylky.

<b>Snímek</b>	<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>
Koeficient zaostření $[\emptyset]$	0,0022	0,0159	<b>0,05</b>	0,0129	0,0026	$7,75 \cdot 10^{-4}$
<b>Snímek</b>	<b>7.</b>	<b>8.</b>	<b>9.</b>	<b>10.</b>	<b>11.</b>	<b>12.</b>
Koef. zaostření $[\emptyset] \cdot 10^{-4}$	4,04	2,25	1,40	0,886	0,808	0,831

## 5.5 Metoda s využitím detektorů hran I.

V metodě se s blokem snímku provedla konvoluce s Laplacianovou maticí, která zní

$$M = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (5.1)$$

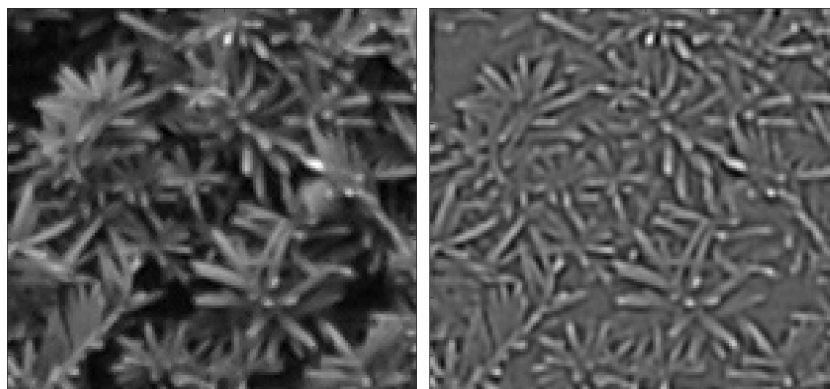
V obrazu se tak zvýraznily hrany a okolí se zalilo stejným odstínem (viz obrázek 5.4). Poté se hodnoty pixelů převedly na druhou mocninu a sečetly se. Tyto koeficienty jsou zaneseny v tabulce 5.5. Čas pro výpočet koeficientů všech bloků byl 0,015 sekundy.

Tab. 5.5: Hodnoty koeficientů vypočítané s pomocí Laplacianovi matice.

<b>Snímek</b>	<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>
Koeficient zaostření $[\emptyset]$	0,553	3,406	<b>89,98</b>	2,831	0,610	0,357
<b>Snímek</b>	<b>7.</b>	<b>8.</b>	<b>9.</b>	<b>10.</b>	<b>11.</b>	<b>12.</b>
Koeficient zaostření $[\emptyset]$	0,297	0,278	0,264	0,2589	0,2594	0,262

## 5.6 Metoda s využitím detektorů hran II.

Metoda vychází ze vzorce 1.6. Provedla se konvoluce obrazu s maticemi  $L_X$  a  $L_Y$ . Výsledky se převedly na kladná čísla pomocí absolutní hodnoty a sečetly. Tyto

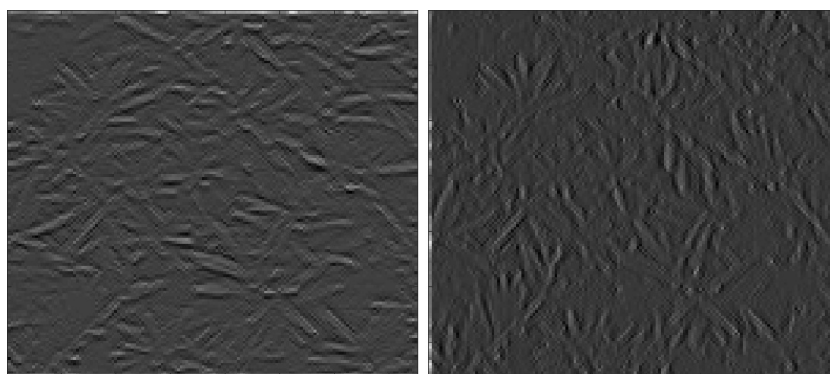


Obr. 5.4: Vlevo originální blok snímku, vpravo blok s implementovanou Laplacianovou maticí.

výsledné koeficienty jsou zaneseny v tabulce 5.6. Výpočet koeficientu u všech bloků trval celkem  $2,797 \cdot 10^{-3}$  sekundy.

Tab. 5.6: Hodnoty koeficientů vypočítané podle vzorce 1.5.

<b>Snímek</b>	<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>
Koeficient zaostření [Ø]	363,9	469,8	<b>1592</b>	436,6	363,5	350,9
<b>Snímek</b>	<b>7.</b>	<b>8.</b>	<b>9.</b>	<b>10.</b>	<b>11.</b>	<b>12.</b>
Koeficient zaostření [Ø]	352,8	354,2	344,1	342,2	341,7	345,8



Obr. 5.5: Vlevo konvoluce s maticí  $L_Y$ , vpravo konvoluce s maticí  $L_X$ .

## 6 Porovnání výsledků analyzovaných metod

Každá výše zmíněná a zároveň otestovaná metoda prezentuje své vypočítané koeficienty ostrosti v přehledné tabulce. Tabulka 6.1 pak souhrnně znázorňuje všechny otestované metody a jejich normalizované koeficienty zaostření. Graf 6.2 tyto hodnoty pro lepší představu vizualizuje.

Z grafu 6.2 či z tabulky 6.1 můžeme vyčíst, že z hlediska odlišitelnosti nejostřejšího obrazu od méně ostrých je na tom nejlépe metoda s využitím detektorů hran I. Následující je metoda se sumou vlnových koeficientů se zhoršením o necelých 10%. Poté následují metody s normalizační odchylkou, s DCT a s detektory hran II, které mají odlišitelnost velmi podobnou. Tyto metody mají zhoršení okolo 5% od předchozí. A nejhorší výsledky má metoda využívající šířku histogramu, kde je odlišnost přes 20% nižší než u předchozích.

Tab. 6.1: Normalizované hodnoty koeficientu ostrosti pro každý snímek v [%].

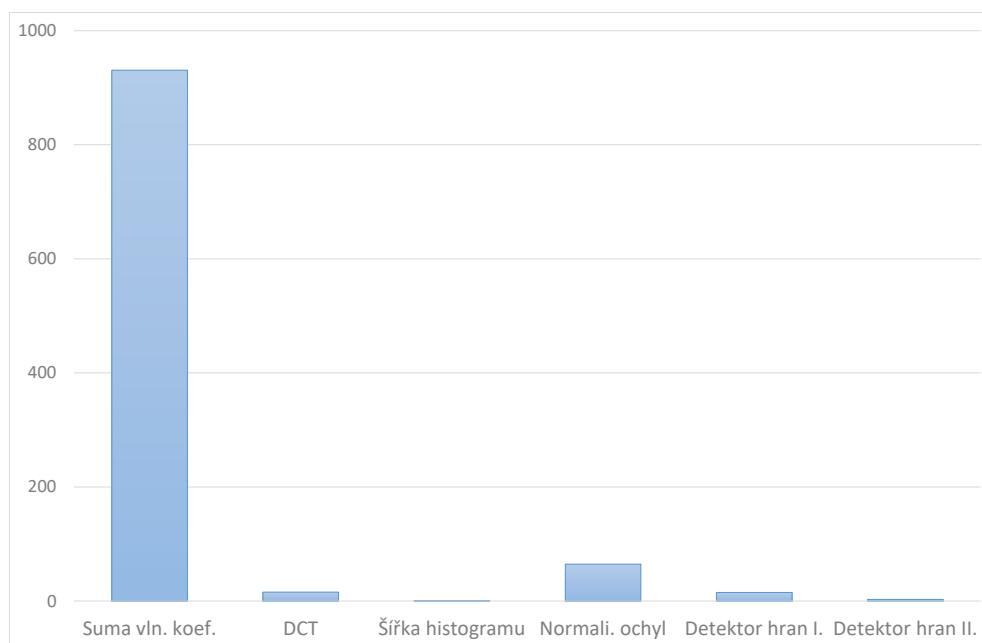
<b>Snímek</b>	<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>
Suma vln.koef.	8,9	24,8	<b>100</b>	21,0	9,4	7,7
DCT	14,1	31,3	<b>100</b>	26,9	14,7	12,2
Šířka histogramu	18,8	58,1	<b>100</b>	46,7	21,8	12,7
Normalizovaná odch.	4,3	31,7	<b>100</b>	25,7	5,2	1,6
Detektor hran I.	0,61	3,78	<b>100</b>	3,15	0,68	0,40
Detektor hran II.	22,9	29,5	<b>100</b>	27,4	22,8	22,0
<b>Snímek</b>	<b>7.</b>	<b>8.</b>	<b>9.</b>	<b>10.</b>	<b>11.</b>	<b>12.</b>
Suma vln.koef.	7,7	7,6	7,7	7,5	7,6	7,7
DCT	11,8	11,5	11,1	10,8	10,8	10,9
Šířka histogramu	10,0	7,4	6,1	6,1	5,2	5,6
Normalizovaná odch.	0,8	0,5	0,3	0,2	0,2	0,2
Detektor hran I.	0,33	0,31	0,29	0,29	0,29	0,29
Detektor hran II.	22,2	22,3	21,6	21,5	21,5	21,7

Časově nejlépe dopadla metoda využívající šířku histogramu s časem 0,676 ms. V řádech desetinásobků času této metody vyšly metody s DCT, s normalizační odchylkou a s detektorem hran I. i II. Časově nejhůř dopadla metoda se sumou vlnových koeficientů s tisícinásobným časem oproti nejrychlejší metodě.

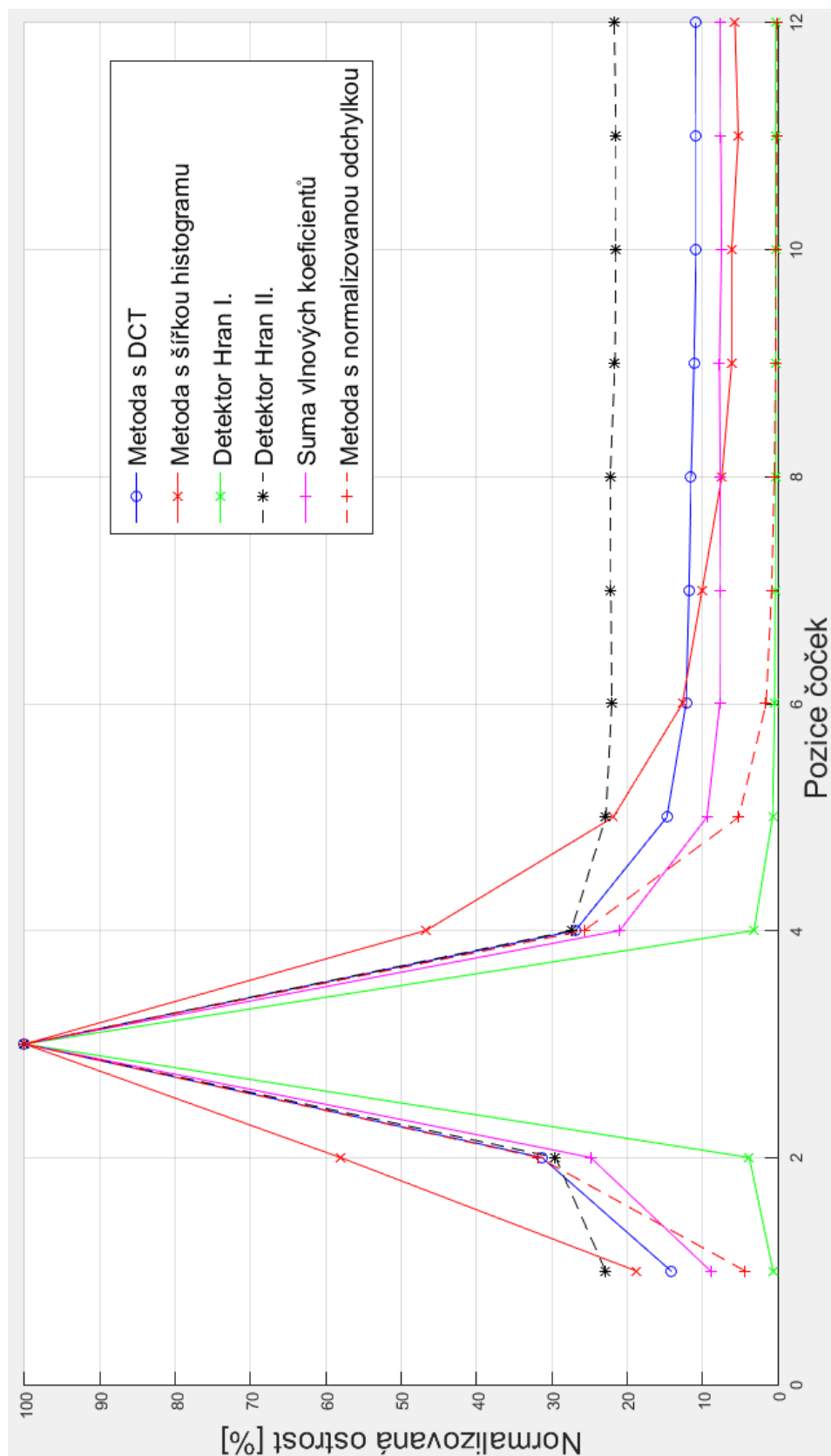
V kapitolách jednotlivých metod jsou doby výpočtů uvedeny. Tabulka 6.2 a graf 6.1 znázorňují hodnoty časů jednotlivých metod.

Tab. 6.2: Hodnoty času, které odpovídají době výpočtů koeficientu zaostření u všech snímků.

Metoda	Hodnota času [ms]
Suma vln.koef.	930,838
DCT	15,587
Šířka histogramu	0,676
Normalizovaná odch.	64,672
Detektor hran I.	15
Detektor hran II.	2,797



Obr. 6.1: Hodnoty časové náročnosti pro všechny otestované metody.



Obr. 6.2: Normalizované hodnoty koeficientu zaostření otestovaných metod každého snímku.

## 7 Program

V této části práce se budeme věnovat popisu výsledného programu. Tato část je rozdělena na dva bloky. V prvním bloku si přiblížíme celkový běh programu a v dalším bloku si popíšeme jednotlivé části programu.

Jak uvádí kapitola 3, tak program vznikl v programu MS Visual Studio 2015 doplněném o knihovnu OpenCV (konkrétně verzi 3.4.5). Podrobnější informace o OpenCV funkcích uvedených v této kapitole nalezneme na stránce [18]. Testování probíhalo na hardwaru uvedeném v téže sekci.

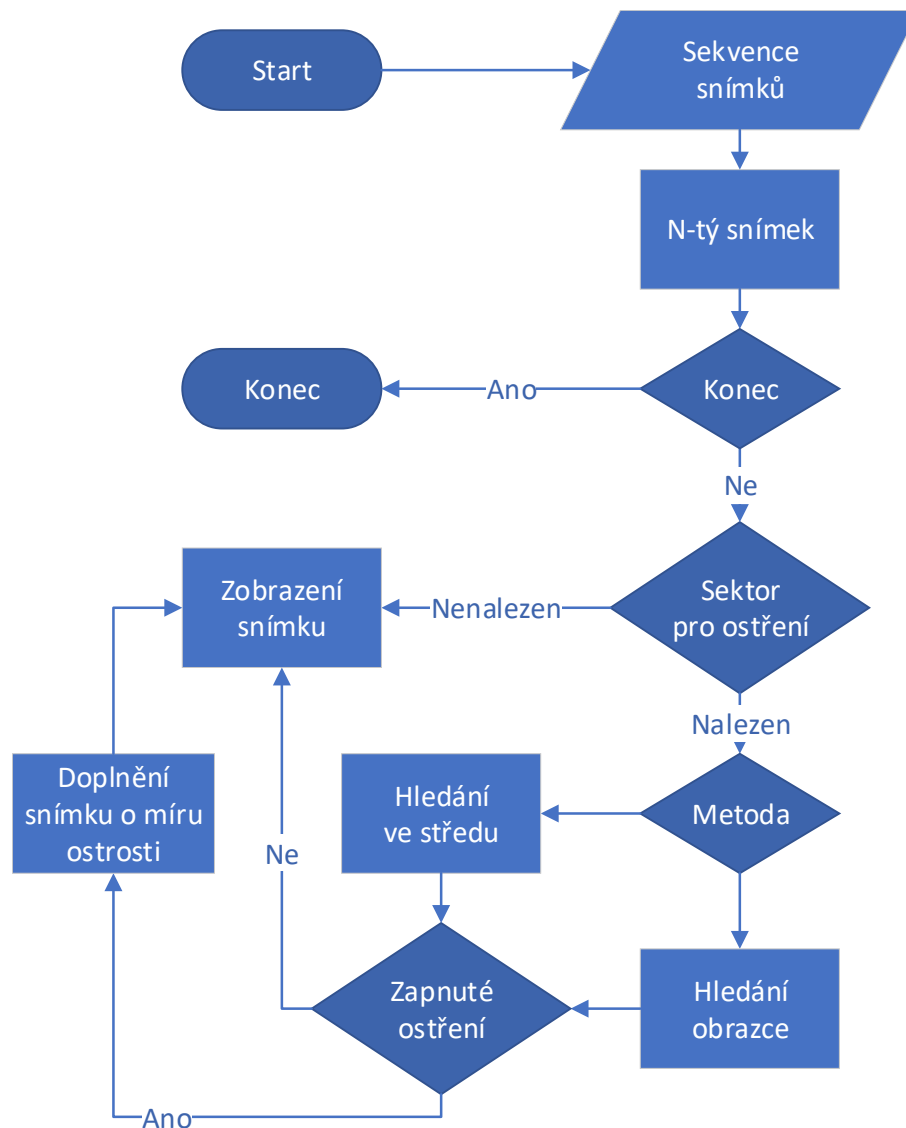
Tento program by se měl využívat právě pro takový typ hardwaru, jaký používáme v této práci. Což znamená, že kamera nijak nekomunikuje s optikou. Jsou pouze mechanicky spojeny. Tudíž se zde nedá využít žádný postup vykonávající plně automatické ostření. Z toho vyplývá, že se ostrost obrazu musí nastavovat manuálně posunováním ostřicí čočky a námi vytvořený software pomůže určit, ve které poloze je obraz nejostřejší.

### 7.1 Obecný popis programu

Vytvořený software odpovídá zadání. Jeho hlavní vlastností je určování míry ostrosti v předem definované oblasti obrazu, který má rozlišení 4K, v reálném čase. Využívá dvě varianty vyhledávání oblasti pro ostření. Jednou variantou je nalezení předem definovaného obrazce ve snímané scéně. Druhá varianta hledá nejlepší místo pro ostření ve střední části obrazu. Pokud nebylo žádné takové místo nalezeno, program vyhodnocuje míru ostrosti ze středu obrazu.

Po nalezení sektoru ostření program může zahájit měření ostrosti. K tomuto procesu byla z výše zmíněných analyzovaných metod vybrána metoda s využitím detektorů hran I. (viz podkapitola 5.5). Metoda byla vybrána pro její schopnost nejlepšího určení maximální míry ostrosti ve srovnání s ostatními testovanými metodami. Dále ani její výpočetní náročnost neomezuje aplikaci pracovat v reálném čase. Jelikož jsme omezeni snímkovou frekvencí kamery (7 FPS) máme i nižší komfortnost doostřování. S vyšší snímkovou rychlostí komfortnost stoupá a snižuje se doba určení maximální hodnoty.

Vývojový diagram na obrázku 7.1 stručně znázorňuje chod programu. Jeho výstupem je sekvence snímků s hodnotou míry ostrosti pro námi vybranou oblast (viz obrázek 7.2).



Obr. 7.1: Vývojový diagram programu.

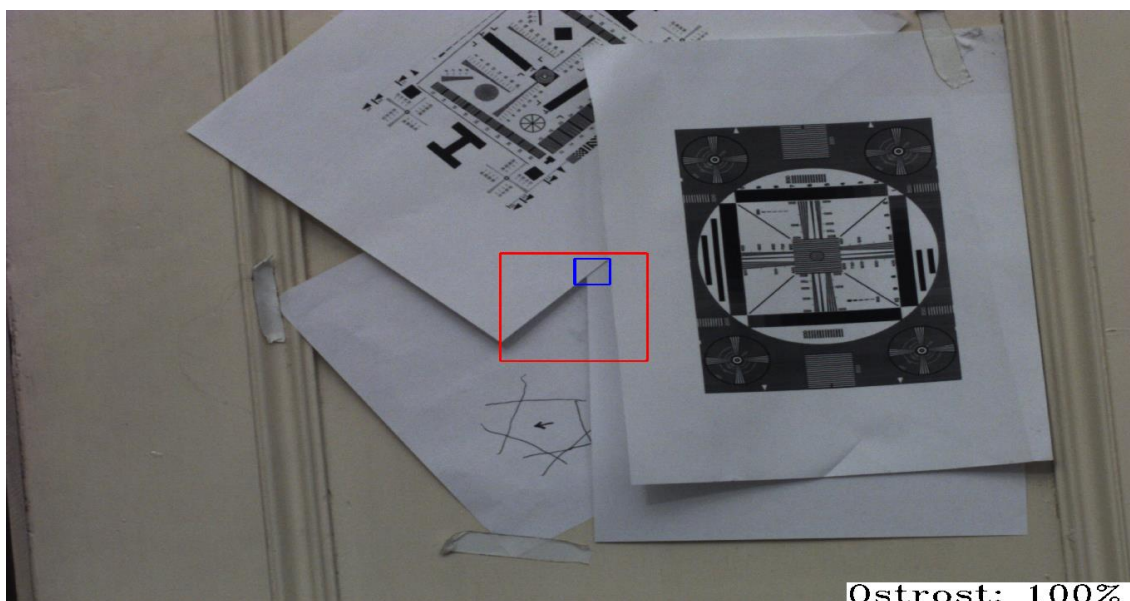
## 7.2 Stručný popis stěžejních částí programu

Zde si přiblížíme stěžejní části programu a uvedeme použité funkce z knihovny OpenCV. Největší pozornost bude mít část s vyhledáváním obrazce ve scéně, vyhledávání nejlepší oblasti pro ostření ve středu obrazu a samotná metoda pro výpočet koeficientu ostrosti.

### 7.2.1 Hledání obrazce ve scéně

Hlavním úkolem této části programu je najít předem definovaný obrazec ve snímané scéně. Když program takový obrazec najde, začne se výpočet ostrosti soustředit na tu část obrazu, ve které se obrazec nachází. Při případné absenci obrazce ve





Obr. 7.2: Snímek kamery, který je doplněný o hodnotu ostrosti, po zpracování programem.

scéně nebo u nesmyslných hodnot souřadnic způsobených chybným detekováním obrazce program zahlásí, že se obrazec nepodařilo najít, a vrátí se zpět.

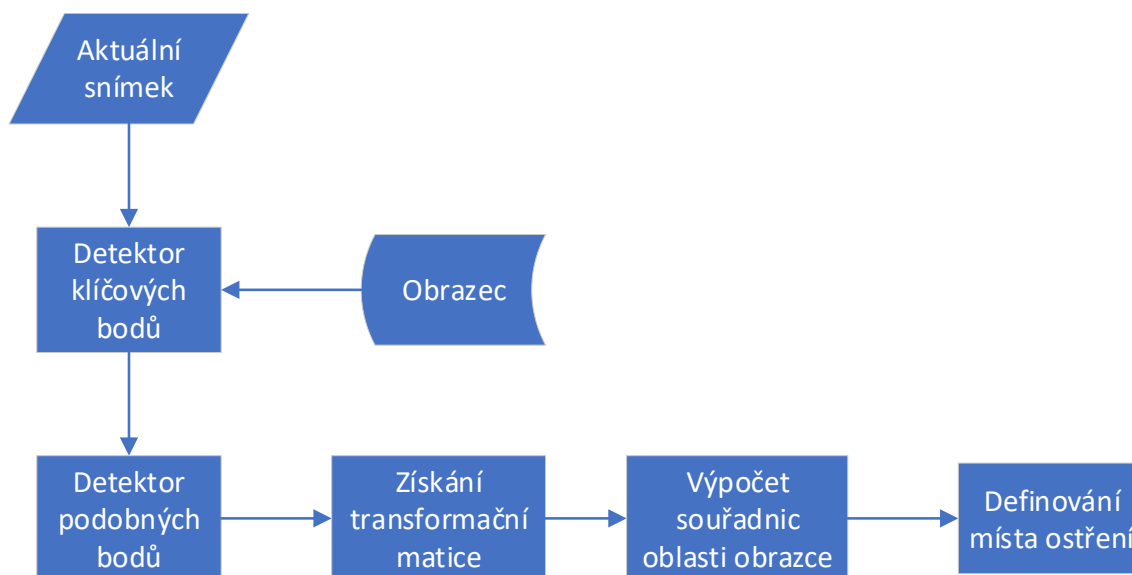
V prvním kroku si program vezme aktuální snímek ze vstupní sekvence snímků. Tento snímek projde filtrovací funkcí, konkrétně *medianBlur()*, která odstraní nežádoucí šum.

V další fázi se pomocí detektoru *SurfFeaturesFinder* ve snímku vyhledají klíčové body. Totéž se provede se snímek, na kterém se nachází obrazec, jenž se ve scéně bude vyhledávat. Funkcí *AffineBestOf2NearestMatcher* se vyberou nejpodobnější body bodům obrazce z bodů snímku zobrazující scénu.

Pomocí těchto vybraných klíčových bodů se vytvoří transformační matice  $M$ , díky které se dopočítají souřadnice výskytu obrazce ve snímku. Pro kontrolu správnosti se nalezený obrazec ohraničí červeným rámečkem (viz obrázek 7.4). Ze souřadnic se vypočítá středový bod, kolem kterého se vytvoří oblast pro měření ostrosti (modrý rámeček).

### 7.2.2 Výběr místa ostření bez obrazce

Začátek funkce je obdobný jako u předchozího postupu. Z aktuálního snímku se tak odstraní nežádoucí šum, ale navíc se vyřízne oblast, ve které chceme hledat neefektivnější místo pro měření. Tato oblast se nachází ve střední části obrazu, z důvodu budoucího použití kamer, které budou snímat tribuny stadionu s diváky na velkou vzdálenost. Zde se předpokládá, že bude kladen důraz na největší ostrost



Obr. 7.3: Vývojový diagram metody určené pro vyhledání obrazce ve scéně.

ve střední části scény.

Vyříznutá oblast se převede pomocí funkce *cvtColor()* z barevného snímku na snímek složený z odstínů šedi. Poté funkce *Canny()* najde hrany. Podle největší koncentrace hran se vybere místo, na které se bude ostřit. Vycházíme z toho, že na hladkém jednobarevném povrchu se ostrost špatně určuje. V případě, že funkce nenajde žádnou hranu, stane se oblastí pro ostření střed obrazu.

### 7.2.3 Měření ostrosti

Jakmile najdeme pomocí jedné z výše zmíněných metod místo, na kterém se bude určovat míra ostrosti, může program přejít k této části. Ta část z každého příchozího snímku vyřízne definovanou oblast. Z každé oblasti se vypočítá pomocí vytvořeného detektoru hran I. koeficient ostrosti.

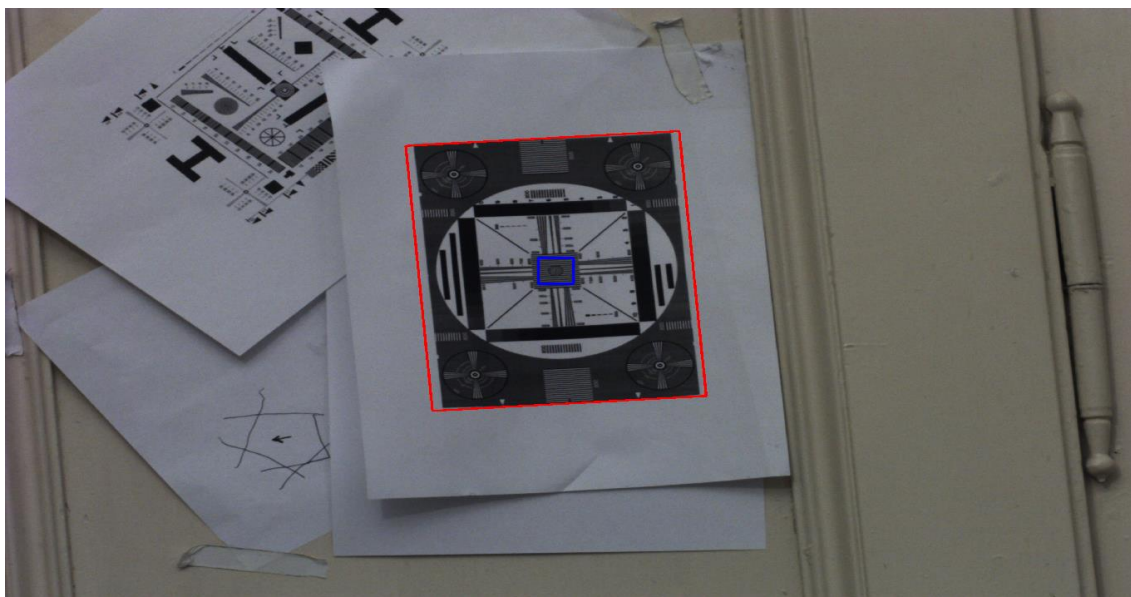
Koeficienty se následně po třech hodnotách průměrují a výsledek se použije k výpočtu míry ostrosti *MO* v procentech. Tu vypočítáme podle vzorce

$$MO = \frac{KO - \min KO}{\max KO - \min KO} \cdot 100, \quad (7.1)$$

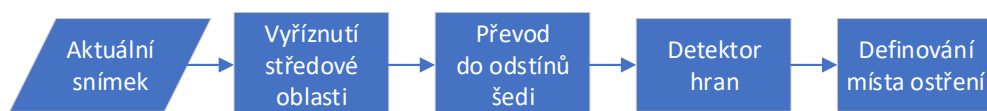
kde *KO* je aktuální hodnota koeficientu ostrosti, *MaxKO* udává největší naměřenou hodnotu koeficientu ostrosti a *minKO* udává nejmenší naměřenou hodnotu.

Odečtením minimální hodnoty od aktuální i maximální docílíme rozsahu od 0 % do 100 %. Aby byla hodnota co nejpřesnější, měl by se proměřit celý rozsah ostřicí čočky, a tím by se správně nastavila maximální a minimální hodnota.

Vybízí se zde i možnost přistupovat k výpočtu maximální hodnoty jinými metodami. Například změřit koeficient ostrosti u různých pozic ostřicí čočky a z těchto



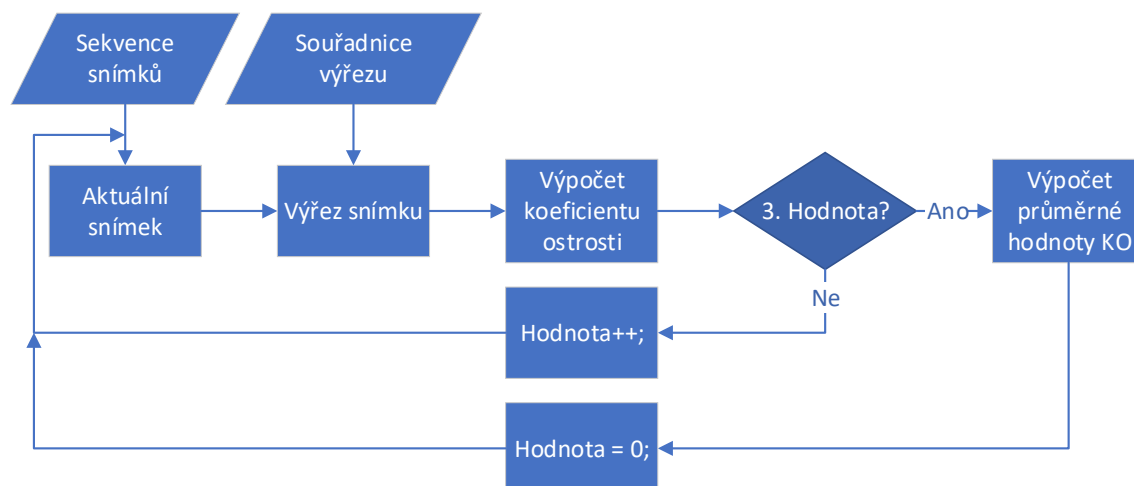
Obr. 7.4: Ukázka nalezení obrazce ve scéně.



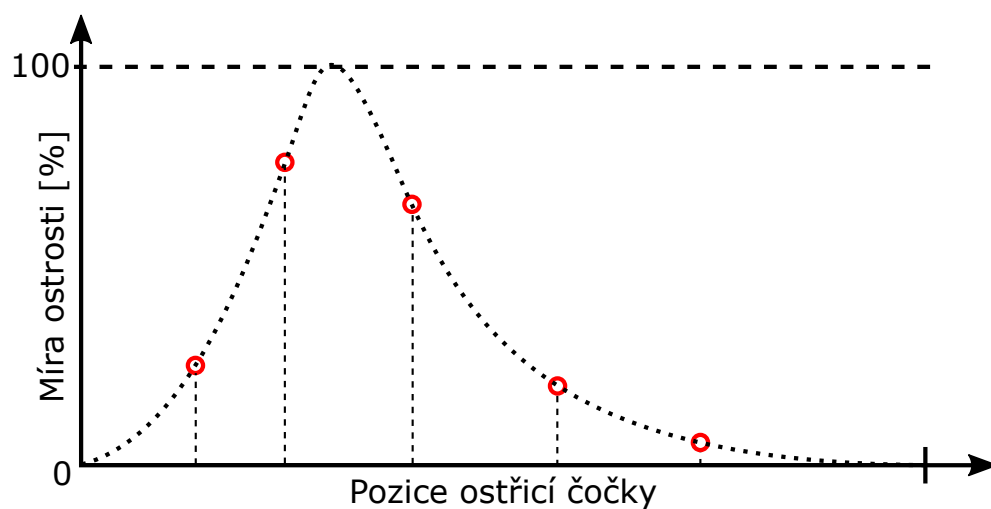
Obr. 7.5: Vývojový diagram metody určené pro vyhledání nejvhodnější oblasti na ostření ve středu obrazu.

hodnot (bodů) získat např. výpočtem aproximační křivky maximální hodnotu koeficientu ostrosti (viz obrázek 7.7).

Tak by bylo maximum určeno mnohem dříve. Tomuto postupu a dalším jemu podobným však brání neznalost pozice čočky. Kamera nemá možnost tuto hodnotu od optiky získat, tudíž neznáme u žádného z bodů souřadnici  $X$ , a tímto se možnost využití této metody vylučuje.



Obr. 7.6: Vývojový diagram metody určené pro vyhledání nejvhodnější oblasti na ostření ve středu obrazu.



Obr. 7.7: Graf znázorňující vytvoření křivky aproximací bodů.

## 8 Testování funkčnosti programu

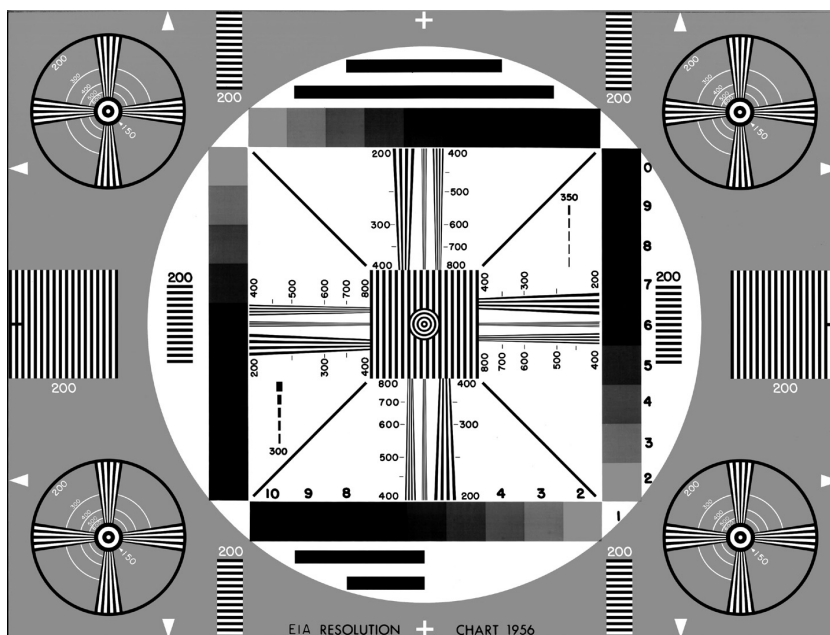
Poslední část se věnuje testování vytvořeného programu. Do programu byly implementovány tři metody určující míru ostrosti. S každou metodou se provedlo šest měření za dobrého osvětlení a šest měření při sníženém osvětlení. Dále se testovalo, do jaké míry ostrosti obrazu dokáže program nalézt pozici obrazce, a do jaké míry ostrosti obrazu dokáže vyhledávat nejvhodnější pozici pro ostření.

### 8.1 Přednastavení programu

V programu se můžou nastavit rozměry oblasti ostření, velikost oblasti ve střední části obrazu, ve které se hledá oblast ostření, a implementuje se obrazec, který má program hledat.

Pro měření byly zvoleny rozměry oblasti ostření  $120 \times 120$  pixelů, zóna pro hledání nejvhodnější oblasti pro měření má rozměry  $500 \times 500$  pixelů. Obrazec, který program ve scéně vyhledává, znázorňuje obrázek 8.1.

Program využíval automatické vyvážení bílé pouze při startu programu. Při měření bylo neměnné z důvodu kvalitativního porovnávání výsledků.



Obr. 8.1: Obrazec, který je programem vyhledáván [19].

## 8.2 Měření maximální míry ostrosti

Pro měření maximálního ostření byly použity tři metody. Konkrétně se jedná o detektor hran I., o metodu s normalizovanou odchylkou a o metodu využívající šířku spektra. Měření všech metod proběhlo v totožné scéně s konstantním osvětlením.

Manuálním ostřením a programem se vždy hledalo maximální zaostření výše definované oblasti umístěné uprostřed obrazu. Při nalezení stoprocentní míry ostrosti se snímek scény uložil. Takto se měření opakovalo šestkrát u každé metody. Po té se ve scéně snížila intenzita osvětlení a měření se opakovalo stejným způsobem.

U každého z výsledných snímků se v oblasti umístěné ve středu obrazu o rozměrech  $500 \times 500$  pixelů vypočítal koeficient ostrosti metodou detektor hran I. Tyto hodnoty obsahuje tabulka 8.1.

Tab. 8.1: Hodnoty koeficientů ostrosti získané z uložených snímků.

<b>Normální osvětlení</b>	<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>
Šířka histogramu	3,41412	3,33642	3,27284	3,45712	3,31355	3,4484
Normalizovaná odch.	3,39018	3,43945	3,35642	3,47736	3,37786	3,40707
Detektor hran I.	3,26757	3,25481	3,30734	3,22112	3,18665	3,28236
<b>Snížené osvětlení</b>	<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>
Šířka histogramu	1,78376	1,78998	1,78691	1,78965	1,80288	1,77688
Normalizovaná odch.	1,7872	1,8041	1,7700	1,79545	1,78561	1,75928
Detektor hran I.	1,79663	1,80848	1,80009	1,80718	1,79665	1,78371

## 8.3 Testování vyhledávání obrazce a vyhledávání nejvhodnějšího místa pro ostření

Měření zde začínalo nalezením 100% míry ostrosti. Následně se začala míra ostrosti po krocích snižovat. V každém kroku byla odečtena míra ostrosti a bylo spuštěno vyhledávání obrazce. Program obrazec přestal nacházet u hodnoty  $45 \% \pm 5 \%$ .

U hledání nejvhodnějšího místa pro ostření na střed obrazu byl zvolen stejný postup. Zde program přestal určovat nejvhodnější místo u  $50 \% \pm 5 \%$ .

Minimální hodnoty míry ostrosti pro hledání v obraze se mohou u jiných scénářů (jiné osvětlení, jiná vzdálenost kamery od místa ostření, změna scény) lišit. Tyto minimální hodnoty, ve kterých program stále dokáže vyhledávat definované místa ve scéně, by mohly být sníženy např. změnou kontrastu snímaného obrazu.

## 8.4 Zhodnocení výsledků

Měření bylo provedeno tak, že se ve stejné scéně, kterou snímá kamera, opakovaně hledala maximální míra zaostření obrazu manuálním posouváním ostřicí čočky ve spolupráci s vytvořeným programem. Kamera byla statická, tudíž se scéna a místo ostření neměnilo. V programu se před každým měřením implementovala jedna ze tří metod určených k výpočtu koeficientu ostrosti. Pro zjištění změn, které přijdou se změnou osvětlení, bylo provedeno totožné měření se sníženou intenzitou světla.

Po nalezení maximální míry zaostření obrazu se u každého měření uložil snímek tohoto obrazu. Poté byl u všech uložených snímků detektorem hran I. vypočten koeficient ostrosti. Použitím stejné metody u všech snímků můžeme určit, která metoda pomáhala určit maximální ostrost nejlépe.

Za normálních světelných podmínek vyšla nejvyšší průměrná hodnota koeficientů ostrosti u metody s normalizovanou odchylkou. Tato hodnota byla 3,41 s odchylkou  $\pm 1,74\%$ . U metody využívající šířku histogramu vyšla průměrná hodnota 3,37 s odchylkou  $\pm 2,67\%$ . Nejnižší hodnotu 3,25 s odchylkou  $\pm 1,82\%$  určoval detektor hran I.

U snížené intenzity světla vyšla u detektoru hran I. nejvyšší průměrná hodnota koeficientu zaostření 1,80 s odchylkou  $\pm 0,68\%$ . U metody využívající šířku histogramu vyšla průměrná hodnota 1,78 s odchylkou  $\pm 0,72\%$ . Nejnižší průměrná hodnota byla 1,78 s odchylkou  $\pm 1,24\%$  u metody s normalizovanou odchylkou.

Z výsledku vidíme, že se průměry koeficientů ostrosti liší pouze o setiny a jejich odchylky se liší o desetiny procent. Dále tímto měřením bylo otestováno, žádná metoda neomezuje program pracovat v reálném čase. Z toho plyne, že by se muselo provést velké množství měření s různými scénáři, aby bylo možné určit, která metoda je pro použití v programu nejlepší. Proto se zde u výběru metody vychází z analýzy metod. Jedná se o detektor hran I., který v analýze vyšel s nejlepšími výsledky u určování maxima a výpočetně program nezatěžuje.

Využít obrazec pro určení místa ostření v obrazu se dá za předpokladu, že oblast, kde se obrazec nachází, je zaostřena alespoň na 50 %. V případě neúspěchu nalezení či absence obrazce v obraze se dá využít funkce pro vyhledávání místa ostření ve středu obrazu. Funkce by měla sama určit nejlepší oblast (pokud se nejedná čistě hypoteticky o jednolitou plochu) pro ostření do 55 % míry ostrosti dané oblasti. Když takové místo nenalezne, tak vybere pro ostření střed obrazu.

## 9 Závěr

V práci byly uvedeny některé z mnoha metod, které vyhodnocují zaostření obrazu v reálném čase. Následně byla většina z těchto metod algoritmicky implementována v programu MATLAB. Pro ideální analyzování metod byla vytvořena sekvence 12 snímků s odlišnou pozicí zaostřovacích čoček. Výsledkem každé metody byl zaostřovací koeficient. Největší koeficient určuje nejzaostřenější obraz. Všechny metody určily správně nejostřejší snímek. S nejpřesnějším vyhodnocením vyšla metoda s detektorem hran I., ale naopak nejrychleji to určila metoda s využitím šířky histogramu.

V další části práce se pojednává o vytvořeném programu, jehož stěžejním bodem je metoda pro výpočet míry ostrosti obrazu. Výpočetní výkon, potřebný k získání hodnoty ostrosti, běh aplikace v reálném čase nijak neomezuje. Tímto jsou splněny hlavní požadavky na software. Program byl vytvořen ve vývojovém prostředí MS Visual Studio 2015, které bylo doplněno knihovnou OpenCV. Výběr oblasti pro zaostření řídí dvě metody. První metoda hledá předem definovaný obrazec ve scéně a v tomto místě program může začít měřit. Druhá metoda vyhledává ve střední části obrazu oblast s největší četností hran. Výstupem programu je procentuální míra zaostření obrazu, které se v reálném čase obnovuje a zobrazuje v pravém dolním rohu snímku.

Měření orientované na určování maximální hodnoty zaostření obrazu ukázalo, že měřené metody se od sebe v průměrovaném maximálním zaostření liší jen v řádu setin. Zároveň všechny metody dovolují programu pracovat v reálném čase. Pro určení nejvhodnější metody by muselo proběhnout velké množství měření s různými scénáři. Proto zde vycházíme z výsledků kapitoly 6 a v programu je implementován detektor hran I.

Program může být využit k doostřování obrazu u aplikací, které používají podobný hardware jako bylo využito v této práci. Nebo samostatně jako doostřovací software kamer, které nemají možnost si řídit optickou soustavu samy.



# Literatura

- [1] LEVOY, Marc, Nora WILLETT a Andrew ADAMS. Autofocus: phase detection. *Stanford Computer Graphics Laboratory* [online]. 2012 [cit. 2019-05-15]. Dostupné z: <https://graphics.stanford.edu/courses/cs178/applets/autofocusPD.html>
- [2] LEVOY, Marc, Nora WILLETT a Andrew ADAMS. Autofocus: contrast detection. *Stanford Computer Graphics Laboratory* [online]. 2012 [cit. 2019-05-15]. Dostupné z: <https://graphics.stanford.edu/courses/cs178/applets/autofocusCD.html>
- [3] GE YANG, B.J. a B.J. NELSON. Wavelet-based autofocusing and unsupervised segmentation of microscopic images. In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* [online]. USA: IEEE, 2003, **3**, s. 2143-2148 [cit. 2018-12-09]. DOI: 10.1109/IROS.2003.1249188. ISBN 0-7803-7860-1.
- [4] PERTUZ, Said, Domenec PUIG a Miguel Angel GARCIA. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition* [online]. Elsevier, 2013, **46**(5), 1415-1432 [cit. 2018-12-09]. DOI: 10.1016/j.patcog.2012.11.011. ISSN 0031-3203.
- [5] NAYAR, S, Y NAKAGAWA a S NAYAR. Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 1994, **16**(8), 824-831 [cit. 2018-12-09]. DOI: 10.1109/34.308479. ISSN 0162-8828. Dostupné z: <http://search.proquest.com/docview/28616906/>
- [6] SANG-YONG LEE, Y., Y. KUMAR, Y. JI-MAN CHO, Y. SANG-WON LEE a Y. SOO-WON KIM. Enhanced Autofocus Algorithm Using Robust Focus Measure and Fuzzy Reasoning. *Circuits and Systems for Video Technology, IEEE Transactions on* [online]. USA: IEEE, 2008, **18**(9), 1237-1246 [cit. 2018-12-08]. DOI: 10.1109/TCSVT.2008.924105. ISSN 1051-8215.
- [7] MATEOS-PÉREZ, José María, Rafael REDONDO, Rodrigo NAVA, et al. Comparative evaluation of autofocus algorithms for a real-time system for automatic detection of Mycobacterium tuberculosis. *Cytometry. Part A: the journal of the International Society for Analytical Cytology* [online]. 2012, **81**(3), 213 [cit. 2018-12-08]. DOI: 10.1002/cyto.a.22020.

- [8] SANTOS, A., C. ORTIZ DE SOL-RZANO, J. J. VAQUERO, J. M. PEÑA, N. MALPICA a F. DEL POZO. Evaluation of autofocus functions in molecular cytogenetic analysis. *Journal of Microscopy* [online]. Oxford, UK: Blackwell Science, 1997, **188**(3), 264-272 [cit. 2018-12-08]. DOI: 10.1046/j.1365-2818.1997.2630819.x. ISSN 0022-2720.
- [9] LOWE, David. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* [online]. Boston: Kluwer Academic Publishers, 2004, **60**(2), 91-110 [cit. 2019-05-05]. DOI: 10.1023/B:VISI.0000029664.99615.94. ISSN 0920-5691. Dostupné z: <https://link-springer-com.ezproxy.lib.vutbr.cz/content/pdf/10.1023%2FB%3AVISI.0000029664.99615.94.pdf>
- [10] KARAMI, E., PRASAD, S. a SHEHATA, M. *Image Matching using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images* [online]. Ithaca: Cornell University Library, arXiv.org, Oct 07, 2017 ProQuest Central; ProQuest Technology Collection [cit. 2019-05-05].
- [11] BAY, H., T. TUYTELAARS a L. VAN GOOL. SURF: Speeded Up Robust Features. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* [online]. Katholieke Universiteit Leuven, 2006, **3951**, s. 404-417 [cit. 2019-05-05]. DOI: 10.1007/11744023\_32. ISBN 3540338322. ISSN 03029743. Dostupné z: [https://link.springer.com/chapter/10.1007%2F11744023\\_32](https://link.springer.com/chapter/10.1007%2F11744023_32)
- [12] RUBLEE, E, V RABAUD, K KONOLIGE a G BRADSKI. ORB: An efficient alternative to SIFT or SURF. In: *2011 International Conference on Computer Vision* [online]. IEEE, 2011, s. 2564-2571 [cit. 2019-05-06]. DOI: 10.1109/ICCV.2011.6126544. ISBN 9781457711015. Dostupné z: <https://ieeexplore.ieee.org/document/6126544>
- [13] JOHNSON, B. *Professional visual studio 2013: selected readings*. 1st edition. Washington: Microsoft Press. Penguin education. ISBN 11-188-3204-3.
- [14] OpenCV. *About* [online]. 2019 [cit. 2019-05-06]. Dostupné z: <https://opencv.org/>
- [15] *MATRIX VISION - mvBlueFOX3 Technical Documentation* [online]. MATRIX VISION, 2019 [cit. 2019-05-10]. Dostupné z: [https://www.matrix-vision.com/manuals/mvBlueFOX3/Appendix\\_CMOS\\_page\\_0.html#mvBC\\_\\_subsubsection\\_sensors\\_CMOS\\_models\\_1100](https://www.matrix-vision.com/manuals/mvBlueFOX3/Appendix_CMOS_page_0.html#mvBC__subsubsection_sensors_CMOS_models_1100)

- [16] *Objektiv Yongnuo 50mm f/1.8 Canon EF* [online]. Hong Kong Yong Nuo Photographic Equipment Co. [cit. 2019-05-10]. Dostupné z: <https://www.yongnuo.cz/yn-50mm-f-18#tb1=1>
- [17] Návod k obsluze DSLR-A300/A350.
- [18] DOXIGEN. *OpenCV* [online]. OpenCV, 2018 [cit. 2019-05-13]. Dostupné z: <https://docs.opencv.org/3.4.5/>
- [19] Focus chart. In: *Gelcompany* [online]. 2019 [cit. 2019-05-15]. Dostupné z: <http://gelcompany.com/wp-content/uploads/2018/09/FocusChart.jpg>

## Seznam symbolů, veličin a zkratek

<b>4K</b>	Obraz o horizontálním rozlišení přibližně 4000 pixelů
<b>DCT</b>	Discrete Cosine Transform – Diskrétní kosinova transformace
<b>DFT</b>	Discrete Fourier Transform – Diskrétní fourierova transformace
<b>DoG</b>	Difference of Gaussian – Gaussův rozdíl
<b>DWT</b>	Discrete Wavelet Transform – Diskrétní vlnová transformace
<b>FPS</b>	Frames Per Second – Snímková frekvence
<b>OpenCV</b>	Open Source Computer Vision Library – Knihovna počítačového vidění s otevřeným zdrojovým kódem
<b>ORB</b>	Oriented FAST and Rotated BRIEF – Orientovaný FAST a otočený BRIEF
<b>SDK</b>	Software Development Kit – Sady pro vývoj softwaru
<b>sRGB</b>	standard Red Green Blue – Standardní červená zelená modrá
<b>SIFT</b>	Scale Invariant Feature Transform – Měřítko invariantních vlastností transformace
<b>SURF</b>	Speeded Up Robust Features – Zrychlené robustní funkce

# Seznam příloh

A	Návod k obsluze programu	49
B	Obsah přiloženého CD	50

## A Návod k obsluze programu

Program je napsán tak, že se obsluhuje pětící kláves. Zmáčknutím klávesy:

- **S** se uloží aktuální snímek do složky *Ulozene\_obrazky*,
- **I** dostaneme aktuální informace o kaměře,
- **O** se začne vyhledávat obrazec ve scéně,
- **M** program spustí vyhledávání nejvhodnější oblasti pro ostření ve střední části obrazu, v případě neúspěšného nalezení takové oblasti se pro ostření vybere střed obrazu,
- **K** se spustí měření procentuální míry ostrosti, za podmínky, že byl nalezen obrazec ve scéně nebo bylo vybráno ostření na střed.

## B Obsah přiloženého CD

Program byl vytvořen a testován ve vývojovém prostředí MS Visual Studio 2015 s přidanou knihovnou OpenCV verze 3.4.5. V tomto vývojovém prostředí se také doporučuje spouštět.

```
/ ..... kořenový adresář přiloženého CD
├── Diplomova_prace_program ..... soubory a zdrojové kódy programu
│   ├── Diplomova_prace ..... soubory a zdrojové kódy programu
│   │   ├── obr ..... obrázky obrazců
│   │   ├── Ulozene_obrazky ..... adresář pro ukládání snímků
│   │   ├── x64
│   │   ├── Diplomova_prace.vcxproj
│   │   ├── Diplomova_prace.vcxproj.filters
│   │   ├── Funkce.h ..... funkce pro výpočet koeficientu ostrosti
│   │   └── Source.cpp ..... kód programu
│   └── x64
│       ├── Diplomova_prace.sln ..... soubor pro spouštění projektu
│       └── Diplomova_prace.VC.db
└── Diplomova_prace.pdf ..... elektronická verze diplomové práce
```